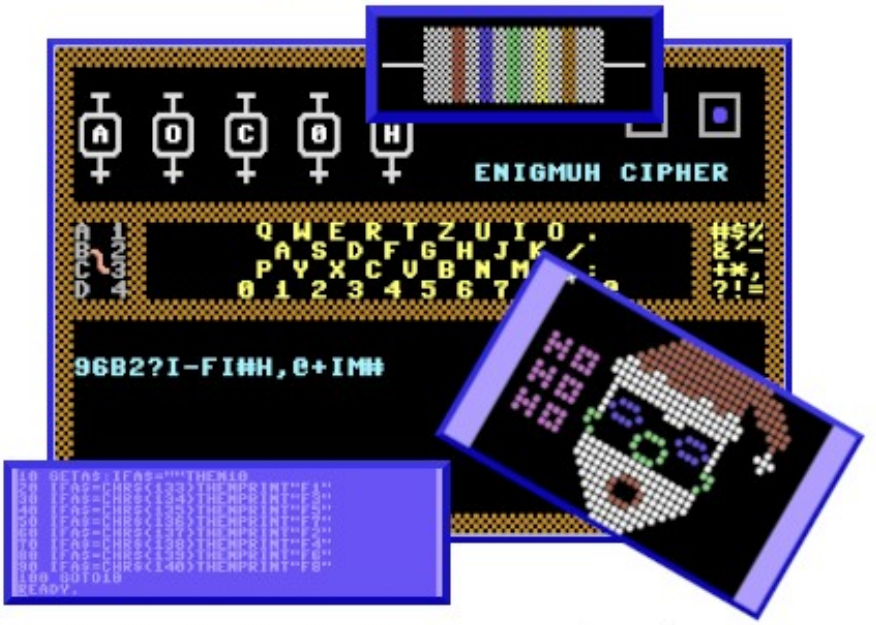
Joes 2nd Book For The C64



By Joseph E. Gordon

Table of Contents

SpeedScript 3.2....7
- Easy Cursor....7
- Fontmaker....7
- Instant 80....7
- Disk Drive Commands....8
- Search (Hunt) and Replace....8
- Copy and Paste....8
- ScriptSave....8
- Search Documents....9
- Spell Checker....10
- Print To Text File....10
- Word Count....11
- Other Speedscript Commands....11

Base Conversions....12
- C128....12
- Decimal to Octal....12
- Universal Base Converter....12

Coding Programs....13
- Coder....13
- Enigma....13
- Enigmuh V2.1....13
- Secret Writing....14
- Secret Noter....14

Databases....17
- Info-Flow 64....17
- Speed File....17
- DataFile 3.3 for C64....18
- DataFile 3.6 for C64....18

Disk Utilities....19
- 1581 Toolkit....19
- 1581 Alphabetizer....19
- Auto Run a Program....19
- Auto Run a Program After Load....20
- Blank Filename....20
- Copy 81....20
- Directory Filer....20
- Directory Manipulator....20
- DirMaster....20
- Disk Doctor....20
- Disk Encoder....21

Disk Magic....21
Disk Scrambler - Encryption....21
Disk Stamper....21
Disk Title Changer....21
Disk Unscratch....21
Dr Jimmy....22
Fast File Copier....22
Fast Load Cartridge....22
File Encryption....22
GCR....22
Kwik Copy....23
Load First File....23
Load From Directory....23
Load Protection....23
Lock Disks....24
Lock Files....24
Menu First....24
Menu Maker....24
Password Sentry....24
Scratch By Filetype....25
Scratch One Character....25
Secure....25
SEQ File Converter....25
SYS Stamper....25
Top Secret....25
Tracks and Sectors....26
Unscratch....26
Using the 1541 Disk Drive....26
Wildcard Directory Listings....27
Eighty Column Screen....28
64 Eighty....28
Left Arrow Functions....28
80 Columns 19XX....29
Screen-80....29
Fonts....30
64 Character Set....30
Mini Font Editor....30
MyC64....30
UltraFont+....31
Games....32
1000 Miler....32

Bard’s Tale I: Tales of the Unknown....32
https://youtu.be/Fs35kuVFpyk....32
Crazy Eights....32
Cribbage Master....32
Godzilla....32
Logic....33
Mancala....33
Silent Service....33
Spy Vs. Spy....33
Yahtzee....33
Zaxxon....33
Misc Programs....34
Family Tree....34
RUN Paint....34
Music....35
Basically Music....35
Music Construction Set....36
Music Keyboard....36
Music Machine....36
Metronome....36
Notepads....37
Crypt Noter....37
Final Cartridge III....37
Notemaker....37
Notepad 64....37
POKEing Around....38
List Directory Without Losing Program....40
Set Cursor Position....40
Programming Utilities....41
64 Searcher....41
Abbreviate Zero....41
Animated Titles....41
Append Two Basic Programs....41
Append Two Basic Programs With Speedscript....42
Automatic Centering....42
Change Ready Prompt....42
Character Fade & Restore....42
Clear Returns & Loops....42
Cold Start....42
Color Lister....43
Colorful Listings....43

Copyright Message....44
Double Quote After Input....44
Dual Screen 64....44
Erase Screen Lines....44
Find Character POKE Codes....44
Free Memory....45
Get Statement Trick....45
Hide Lines In LIST....45
Highlighted Menu....45
Indenting....45
INPUT Default Value....45
Inputting Commas....46
INPUT Without Question Mark....46
Joystick....47
Line Freeze....47
List Protection....47
Load Disk Menu From Program....48
MetaBASIC....48
PEEK(197) Table....50
PRINT@....51
Programming Function Keys....51
Quote Mode....51
Random Numbers....51
REDIM'D ARRAY ERROR....52
Recover After NEW Command....52
Renumber....52
Screen Memory Location Finder....52
Screen Scroll....53
Screen Shake....53
Sequential File Editor....53
Show Directory Without Losing Program....53
Split 64....53
Start/End Address....54
Text Centering....54
Time Variable....55
Trace....55
Triple 64....55
Universal Input....55
Spreadsheets....56
CalcAid 64....56
SpeedCalc....57

Export Cell Blocks....57
Keyboard Controls....57
Functions....58
Notes:....58
SwiftCalc 128....58
SYSops....59
WAIT for it....60
Windows....61
64 Windows....61
Pop-Up Menus....61
Quick Windows....61
Window Wizard....61
Enigmuh V2.1 Program Listing....62
Keyfinder....87
Errata....88

For: Chrislynn, Lucas, Anavianna, Braiden, Landon, Bryleigh, River, Maximilian, Blaine, Emerson, Felix and Ofelia.

SpeedScript 3.2

SpeedScript is a word processor for the Commodore 64. Version 3.2 was published in Compute's Gazette, May 1987.

Easy Cursor

The Compute's Gazette disk, November 1989, has an update for SpeedScript 3.0+ called Easy Cursor. This changes the function of the up and down cursors to move line by line instead of sentence by sentence. The second thing this patch does is correct the overwrite errors by deleting the old saved document and saving the new document instead of just overwriting the old document. Use the following steps to install this update:

```
LOAD"SPEEDSCRIPT",8
LOAD"EASYCURSOR.ML",8,1
SYS10431
SAVE"SPEEDSCRIPTEC",8
```

Fontmaker

SpeedScript Fontmaker can be typed in from Compute! magazine, January 1986. It allows you to use fonts you have created with UltraFont, Compute's Gazette July 1984, or UltraFont+, Compute's Gazette September 1986, in SpeedScript.

Instant 80

Compute's Gazette, December 1987, has the "Instant 80" update allowing and 80 column preview for SpeedScript documents. Use (CTRL+SHIFT+P) then 'S' for screen. Warning: Any key combination in VICE 3.3 using SHIFT and 'P' will cause the emulator to permanently pause until the program is restarted.

Disk Drive Commands

```
F7                              - Load a document
F8                              - Save a document
CTRL+4                          - View disk directory

Use CTRL+Up Arrow (PgDn)to send these commands to the disk drive.

@:filename                      - Overwrite document
s:filename                      - Scratch (delete) file
r:newname=oldname               - Rename a document
c:backupname=originalname       - Copy a document
```

Search (Hunt) and Replace

The hunt/replace feature is comparable to find/replace in Word.

```
SHIFT+CTRL+H                    - Enter phrase to search for
CTRL+H                          - Search for phrase
SHIFT+CTRL+J                    - Enter replacement phrase
CTRL+J                          - Replace the phrase
CTRL+G                          - Find & replace in entire document
```

Copy and Paste

To move text, delete or erase the text and move the cursor where you want the text to appear and restore it. To copy and paste: delete and restore it. Then move the cursor where you want the copied text to start and restore it there also.

```
CTRL+D                          - Delete (Left of cursor)
CTRL+E                          - Erase (Right of cursor)
CTRL+R                          - Restore last deleted/erased
```

ScriptSave

ScriptSave is a utility for SpeedScript that will automatically save your document every ten minutes. This is a type-in program from the book, “Compute’s SpeedScript” available at archive.org. It must be on the same disk as the SpeedScript program. Change line 30 in ScriptSave from ‘SS3’ to the name of the SpeedScript program on the disk. Put ScriptSave on the disk before SpeedScript and use `LOAD"*",8:{SHIFT+RUN/STOP}` to load the program. Use document file names of 14 character or less and ScriptSave will add a two digit version number to the beginning of the filename.

After you save the final version of your document, you can delete all the old versions from within SpeedScript by using:

```
{CTRL+Up Arrow} s:??filename
```

Search Documents

Use the “SpeedSearch” program from Compute’s Gazette May 1987 disk issue to search all the Speedscript 3.x documents on a disk for a word or phrase. If you have a modified version of Speedscript, like the Instant 80 patch, you will have to know the high and low byte of the documents starting address. You can use MetaBASIC from the February 1987 issue of Compute’s Gazette with the following command:

```
START “NAME-OF-DOCUMENT”
```

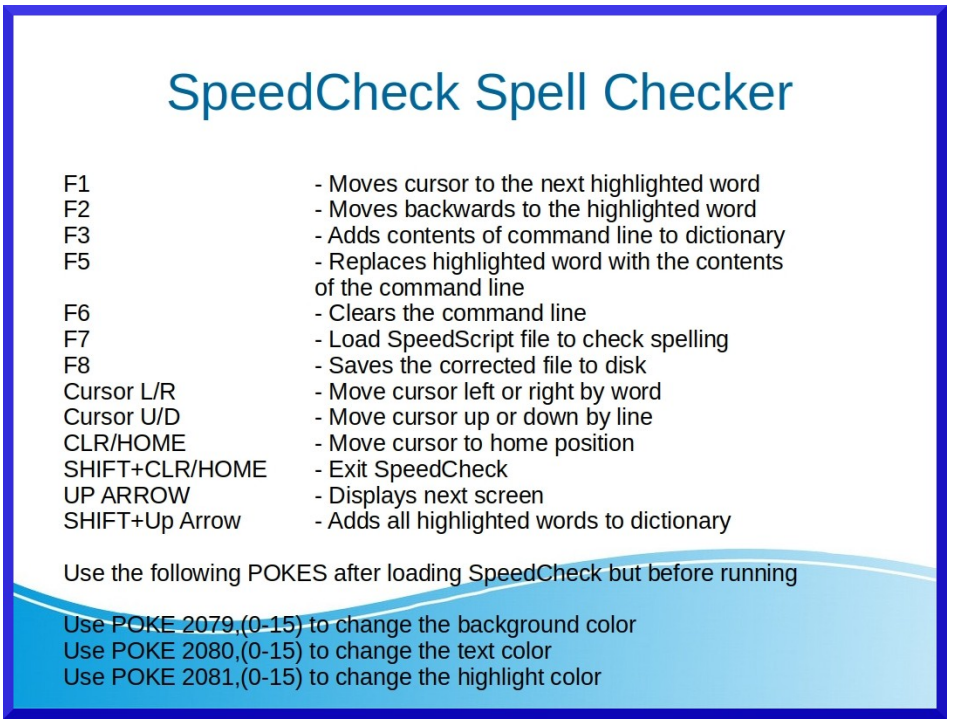
For me, this returned a value of 10240 as the starting address of all documents created with Speedscript modified with the ‘Instant 80’ patch. Next, you will have to find the high and low bytes of the starting address using my Hi-Lo Byte Finder program:

```
10 REM FIND THE HIGH AND LOW BYTE
20 REM OF A STARTING ADDRESS
30 INPUT “MEMORY ADDRESS”;AD
40 HB=INT(AD/256):LB=AD-(HB*256)
50 PRINT”HI BYTE=”;HB
60 PRINT”LO BYTE=”;LB
70 END
```

Once you have these numbers, load SpeedSearch but don’t run it yet. Type “`POKE 2534,LB`” and “`POKE2542,HB`”. Now run SpeedSearch. If it works correctly you can save it to the disk with your documents and it will remember the starting address.

Spell Checker

SpeedCheck is the only spell checker for SpeedScript 3.2 and is in Compute's December 1985 disk, available at archive.org. It comes with a 2,000 word dictionary but allows you to add new words. However, it only checks the first five letters. The commands are:


SpeedCheck Spell Checker
F1 - Moves cursor to the next highlighted word
F2 - Moves backwards to the highlighted word
F3 - Adds contents of command line to dictionary
F5 - Replaces highlighted word with the contents of the command line
F6 - Clears the command line
F7 - Load SpeedScript file to check spelling
F8 - Saves the corrected file to disk
Cursor L/R - Move cursor left or right by word
Cursor U/D - Move cursor up or down by line
CLR/HOME - Move cursor to home position
SHIFT+CLR/HOME - Exit SpeedCheck
UP ARROW - Displays next screen
SHIFT+Up Arrow - Adds all highlighted words to dictionary
Use the following POKES after loading SpeedCheck but before running
Use POKE 2079,(0-15) to change the background color
Use POKE 2080,(0-15) to change the text color
Use POKE 2081,(0-15) to change the highlight color

Print To Text File

You can print your Speedscript document to a text in VICE 3.3 by going to Settings/Settings/Peripheral Devices/Printer Settings and choosing Printer #4, File system access, Enable IEC device, ASCII, Text, Output device #1 and Printer text output device #1 "print.dump". Then press `CTRL+P` to print.

Word Count

Use the ScriptRead program from Compute's Gazette Disk issue May 1987 to get the word count of a Speedscript document. You may have to press "M" for the menu then press "T" or "B" to change the text/border color. When you choose "R" to read a document, the word count will be at the bottom of the text. Other options include:

E – Read the disk error channel
S – Scratch (Delete) filename
D – Show disk directory
Q – Quit directory listing/reading file
CTRL – Slow directory listing
RUN/STOP – Exit

Other Speedscript Commands

F1	Next Word
F2	Previous Word
F3	Next Sentence
F4	Previous Sentence
F5	Next Paragraph
F6	Previous Paragraph
CTRL+B	Change Border Color
CTRL+I	Insert Mode
CTRL+K	Kill (Empty) Buffer
CTRL+L	Change Cursor/Text Color
CTRL+S	Cursor Home
CTRL+Z	Go To End Of Documents
CTRL+=	Display Free Memory
CLR/HOME	Go To Top Of Screen
	Hold To Go To Top Of Text
SHIFT+INST/DEL	Insert Space
RUN/STOP	5 Space Indent
RESTORE	Exit Speedscript
SHIFT+CLR/HOME	Erase All

Base Conversions

C128

VICE comes with a Commodore 128 emulator that has a built converter. Type "MONITOR" at the prompt and enter any number preceded by a code identifying it's base: Decimal (+), Hexadecimal ($), Octal (&) and Binary (%). The monitor will convert that number to all other bases. Enter "X" to exit the monitor.

Decimal to Octal

I wrote the following short programs to convert decimal to octal and octal to decimal because all the conversion programs I found to download or type in from magazines only do binary, decimal and hexadecimal.

```
5 REM ***** OCTAL TO DECIMAL *****
10 N=0:D=0:N$="":F=0:CH$=""
20 INPUT"OCTAL#";N$
30 FORX=0TOLEN(N$)-1
40 CH$=MID$(N$,LEN(N$)-X,1)
50 F=VAL(CH$)
60 N=N+F*8↑X
70 NEXTX
80 PRINTN
90 END
```

```
1600 REM ***** DECIMAL TO OCTAL *****
1605 REM * MAX DECIMAL = 65535 *
1606 REM * MAX OCTAL   = 177777 *
1610 INPUT"DECIMAL#";N$
1620 D=VAL(N$):MD=1:N=0:N$=""
1630 IF 8↑ND>D+1 THEN 1650
1640 ND=ND+1:GOTO1630
1650 FOR X=NDTO1STEP-1
1660 C=INT(D/8↑(X-1))
1670 D=D-C*8↑(X-1)
1680 IFC<8THENN$=N$+STR$(C)
1690 NEXTX
1700 FORX=1TOLEN(N$)
1705 IFMID$(N$,X,1)=" "THEN1720
1710 IFMID$(N$,X,1)<>" "THEND$=D$+MID$(N
$,X,1)
1720 NEXTX:N$=D$:PRINTD$
```

Universal Base Converter

I found a converter in the book, "Tips & Tricks For Commodore Computers", that will convert numbers between base 2 and 36. It is available at archive.org.

Coding Programs

Coder

"Coder" is a message encryption program from the book, "Commodore 64 Programs for the Home", available at archive.org. I have modified the program to print the coded message to a text file in Ubuntu if you have set up VICE to print to a file as described in the Speedscript 3.2 section of this book. Below are the lines I changed or created:

```
120  NEXT I:PRINT"{CLR/HOME}{CYAN}"

1165 OPEN4,4:CMD4
1191 PRINT#4
1195 CLOSE4
1196 FOR I=1 TO I0:PRINTM1$(I):NEXT I
```

Enigma

There is an encryption program called Enigma available on LoadStar Disk 37 and a type in version in the book, "Commodore 64 Fun and Games: Volume 2" available at archive.org.

Enigmuh V2.1

For this version of Enigmuh, I have sped up the screen drawing and added a print to text file feature. When you have completed coding a message, press the F1 key to start the process. The text file will be called 'print.dump' and will be in your Ubuntu home directory. Use the procedure in the Speedscript section to enable printing.

There are two ways to send the coded message. The first is to take a screenshot of the message and send it to someone. The second way is to hide the text message inside of a picture file. I have two YouTube videos that show how to do this:

https://youtu.be/-5Nqc7Bl9PQ
https://youtu.be/LRyYjajt0SA

I have further sped up the operation of Enigmuh by compiling it with Basic Boss, which is the only compiler that worked. This version is available for download at: https://archive.org/details/c-64-enigmuh-coder-v-2.1-compiled. You can speed up the program even more by running the VICE Emulator at 200% speed by changing the following setting:

settings/settings/machine settings/speed settings/speed 200%

Secret Writing

"Your Commodore" published an encryption program is their November 1988 issue called "Secret Writing" and is based on the Caeser Cipher.

Secret Noter

Secret Noter is a program I wrote to read text written on the screen, encrypting it, and saving to disk in a sequential file. I incorporated a small program from a Commodore magazine that saves the screen to memory then restores it when called. To use the program, load it from the disk. You may now enter any text on the screen, but don't hit the RETURN key. The program on saves text from the first 24 rows allowing you to type RUN on the 25th line and press the RETURN key. You will be asked for a three digit encryption code a name for message to save to disk. To read an encrypted message, just load and run the program, enter the name of the file and the three digit code.

```
10 X=0:Y=0:CD=0:NM$="":A$="":PK=0
20 POKE53280,6:POKE646,13:POKE53281,0
30 REM EASY WINDOWS - JEREMY HUBBLE – LINES 40-160
40 REM SYS828 TO SAVE SCREEN
50 REM SYS858 TO RESTORE SCREEN
60 FOR J=828TO887:READK:POKEJ,K
70 CS=CS+K:NEXT
80 IF CS<>7006 THEN PRINT "ERROR":STOP
90 DATA160,000,185,000,004,153,000,192
100 DATA185,000,005,153,000,193,185,000
```

```
110 DATA006,153,000,194,185,232,006,153
120 DATA232,194,200,208,229,096,160,000
130 DATA185,000,192,153,000,004,185,000
140 DATA193,153,000,005,185,000,194,153
150 DATA000,006,185,232,194,153,232,006
160 DATA200,208,229,096
170 SYS828:PRINTCHR$(147)
180 PRINT""
190 INPUT"(C)REATE OR (R)EAD NOTE";IN$
200 IF IN$="R"THEN390
210 INPUT"ENTER NAME OF FILE";NM$
220 INPUT"ENTER 3 DIGIT CODE";CD
230 IF CD<0 OR CD>999 THEN220
240 Y=1
250 SYS858
260 OPEN 8,8,8,"@0:"+NM$+",W,S"
270 FORX=1024TO1983
280 PK=PEEK(X)+CD
290 PRINT#8,PK
300 IF Y=1 THEN CD=CD+7
310 IF Y=2 THEN CD=CD+3
320 IF Y=3 THEN CD=CD+12
330 IF Y=4 THEN CD=CD-13
340 Y=Y+1:IF Y=5 THEN Y=1
350 NEXTX
360 CLOSE8
370 END
380 SYS828:PRINTCHR$(147)
390 PRINT""
400 INPUT"ENTER NAME OF FILE";NM$
410 INPUT"ENTER 3 DIGIT CODE";CD
420 IF CD<0 OR CD>999 THEN410
430 Y=1
440 SYS858
450 PRINTCHR$(147)
```

```
460 OPEN 8,8,8,NM$+",R,S"
470 FORX=1024TO1983
480 INPUT#8,PK
490 POKEX,PK-CD
500 IF Y=1 THEN CD=CD+7
510 IF Y=2 THEN CD=CD+3
520 IF Y=3 THEN CD=CD+12
530 IF Y=4 THEN CD=CD-13
540 Y=Y+1:IF Y=5 THEN Y=1
550 NEXTX:CLOSE8
560 PRINT"{HOME}{DOWN 24}MESSAGE COMPLETE. PRESS ANY KEY.
{UP 2}"
570 GETM$:IFM$=""THEN570
580 END
```

https://archive.org/details/@josephgordon/uploads

Databases

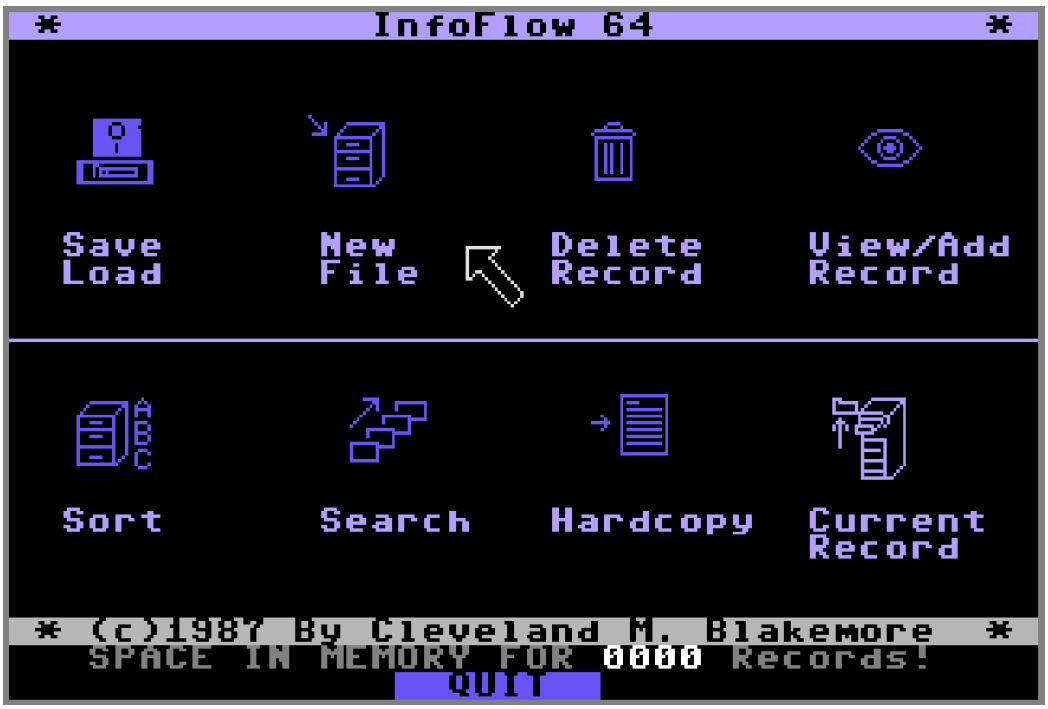
Info-Flow 64

Info-Flow 64 is an icon driven database for the Commodore 64 first published in the January 1988 issue of Ahoy! Magazine and uses a joystick plugged into port 2. I have created a video on how to use this database. It can be viewed here:

https://youtu.be/j4X8rVlxI1g

Speed File

Speed File was published in Compute's Gazette, April 1988. It allows you to create custom screens for entry and viewing. Other features include: Add, Delete, Edit and Sort.



DataFile 3.3 for C64

From the ReRUN Productivity Pak I. Change the value in line 23 from 30000 to 60000 for C128 use.

DataFile 3.6 for C64

From the ReRUN Productivity Pak II. Change the value in line 22 from 29500 to 59000 for C128 use.

Disk Utilities

1581 Toolkit

The VICE emulator has an option for using the 1581 disk drive, which has a lot more storage space than the 1541. The toolkit was published in Commodore Disk User, June 1990, Issue 20. The 1581 Toolkit has the following capabilities:

Fast Data Copier
Fast File Copier
Directory Editor
Track/Sector Editor
Pattern Searcher
Create Partition
Fast Formatter
Error Scanner
Fast Loader

1581 Alphabetizer

This program published in Compute's Gazette, February 1989, allows you to rearrange the directory any way you want. It's main function is to put the files in order alphabetically. This comes in handy because the 1581 has over four times the available storage space and can handle nearly 300 files.

Auto Run a Program

To make your basic programs auto run after loading add the following line to your basic program:

```
0 POKE770,131:POKE771,164
```

Next, enter the following command:

```
PRINT"{CLR/HOME}":POKE770,113:POKE771,168:POKE43,0:
POKE44,3:POKE157,0:SAVE"filename",8
```

Finally, power cycle the C64 and load your file in the following manner:

```
LOAD"filename",8,1
```

Auto Run a Program After Load

```
LOAD"PROGRAM_NAME",8:{SHIFT}{RUN/STOP}
```

Blank Filename

To create a blank (invisible) filename that is hard to delete, add a colon at the end of the name. Use the colon when you need to load the program.

```
SAVE"FILENAME:",8
```

Copy 81

A file copier published in Compute's Gazette, November 1989, for the 1581 disk drive.

Directory Filer

A directory utility from Compute's Gazette, April 1986 Issue 34, that is very similar to Directory Manipulator.

Directory Manipulator

From Ahoy! Magazine, December 1985 (available at archive.org). Directory Manipulator allows you to alphabetize the directory, insert a blank space, insert a dashed line, insert a remark, swap items or delete items from the directory.

DirMaster

DirMaster is a free Windows program for manipulating Commodore disks and can also be installed in Linux using WINE.

Disk Doctor

This program is available on the Re-Run Disk, Fall 1985, March 1985, at archive.org. And is capable of unscratching files.

Disk Encoder

This program from Compute's Gazette, November 1985, will encrypt selected files on the disk. The encoder reads the disks and asks you to select the files you want to encrypt. Next, it asks for a code of ten characters or less then asks for a second code. This second code must have fewer characters than the first.

Disk Magic

With Disk Magic from Compute's October 1987 disk magazine, you can move, lock, delete and unscratch files. You can also change file names and alphabetize the directory.

Disk Scrambler - Encryption

Encrypt specific tracks, blocks (sectors) or the entire 1541 disk. From the August 1989 issue (59) of "Your Commodore" magazine.

https://archive.org/details/disc-scrambler

Disk Stamper

With Disk Stamper from Run magazine's August 1989 disk issue, you can write a secret message on a disk that can only be read with the Disk Stamper program.

Commodore Magazine published their own version of Disk Stamper in Issue 8, August 1987. It can't be deleted, removed by the verify command and doesn't show up in the directory listing.

Disk Title Changer

To change the title of a disk, use Disk Title Changer from the July 1985 edition of Compute's Gazette. Final Cartridge III can also change the title of a disk.

Disk Unscratch

This program will restore deleted files (unscratch) and was published in Your Commodore Magazine Issue 8, May 1985.

Dr Jimmy

Doctor Jimmy will cause your program listings to show up as blank lines by inserting a zero byte after the line number. It was published in Your Commodore Magazine Issue 10, June 1985.

Fast File Copier

From Compute's Gazette, September 1986, Issue 39. This program copies, scratches, renames, formats and validates disks.

Fast Load Cartridge

With Fast Load, you can copy an entire disk, format, copy a file, delete a file, lock/unlock a file and rename a file/disk.

File Encryption

File Encrypter from Run magazine, May 1991, is capable of encrypting most basic files and documents except GEOS files.

GCR

Group Code Recording is the formatting method used by the 1541/1571 drives to write information on the disk.

DEC	HEX	Binary	GCR
0	0	0000	01010
1	1	0001	01011
2	2	0010	10010
3	3	0011	10011
4	4	0100	01110
5	5	0101	01111
6	6	0110	10110
7	7	0111	10111
8	8	1000	01001
9	9	1001	11001
10	A	1010	11010
11	B	1011	11011
12	C	1100	01101
13	D	1101	11101
14	E	1110	11110
15	F	1111	10101

Kwik Copy

Available from commodore.software, Kwik Copy allows you to copy individual files from one disk to another. Also available at:

https://archive.org/details/d64_Kwik_Load_1984_Diskmasters

Load First File

To load the first file on the disk type `LOAD"*",8`. However, if you have previously loaded a program from the current disk, that command will load the other program instead of the first program. To always load the first program, use: `LOAD":*",8`.

Load From Directory

To load a program from the directory listing, type LOAD in the first four spaces to the left of the program name. Next, move to the space just after the quote at the end of the file name and type `",8:"` and hit `RETURN`.

Load Protection

To help stop someone from loading a program from the disk, save the program with at least one `SHIFT+SPACE` as the first character of the filename. Be sure to do the same when loading the program.

You can also save the file as, `SAVE"filename"+CHR$(34),8`. The program will not load unless the `+CHR$(34)` is added to the `LOAD` command.

A third method is to use two `CHR$(31)` codes preceding the filename when saving, `SAVE CHR$(31)+CHR$(31)+"filename",8`. Now load as, `LOAD "??filename",8.`

To make your basic program un-loadable, save it as a sequential or user file: `SAVE"filename,S",8` or `SAVE"filename,U",8`. In order to load these programs you must include the S or U in the load statement:

`LOAD"filename,S",8` or `LOAD"filename,U",8`

Lock Disks

Locking a C64 disk is the same as write protecting it. Files are now read only, nothing can be written to or deleted from the disk. “Disk Lock” is a type-in program from the February 1985 edition of Compute’s Gazette.

Lock Files

Locking files on a C64 disk prevents them from being scratched (deleted). This can easily be done with the Epyx Fast Load cartridge. Locked files can be identified by the “<” symbol to the right of the program type in the directory listing. To lock a file, insert the Fast Load cartridge and enter the British pound symbol (“\” on a keyboard using VICE emulator). Enter “F” to open the File Utility and choose option “E” to lock a file or option “F” to unlock a file.

Menu First

To ensure that your disk menu program is the first program on the disk, you can create a dummy program and save it. After all the programs you want are on the disk, you can delete your dummy program and create your menu. When you save it to the disk, it will be the first program.

Menu Maker

Use Turbo Menu Maker 64 from https://commodore.software to create a nice, bootable menu for your disks and make it the first program. Turbo Menu Maker lets you choose which directory entries will be included in the menu.

Your Commodore Magazine Issue 29, February 1987, has a type-in version and is available at archive.org.

Password Sentry

Password Sentry, published by Commodore Magazine in Issue 4, April 1987, changes a file to a series of random numbers. The program can only be restored with the proper password.

Scratch By Filetype

You can delete all Program, SEQ or User files on a disk by substituting P,S or U for 'X' in the example below:

```
OPEN 15,8,15,"S0:*=X":CLOSE15
```

Scratch One Character

Can't scratch a single character filename?

```
OPEN15,8,15,"S:?":CLOSE15
```

Secure

This is a program from Commodore Disk User, January 1991, Issue 27. Secure prevents your program from being listed. To use it, just load Secure and load your basic program. Now add the following lines to your basic program:

```
0 REM
1 POKE774,226:POKE775,252
```

Now enter `SYS49152` and save your protected program.

SEQ File Converter

This little program, published in RUN magazine as tip $5D3, converts sequential files to PRG. This will allow you to open it with SpeedScript. You could also try tip $4FF:

```
OPEN8,8,8,"FILENAME,P,W":CMD8:LIST:PRINT#8:CLOSE8
```

SYS Stamper

This program from Compute's Gazette, July 1988, will display the starting address of file in the disk directory.

Top Secret

Another file encryption program from Compute's Gazette, November 1987. Load Top Secret and remember the SYS address you are given. Load the basic program you want to encrypt and type in the `SYS ADDRESS` you were given. Enter a password consisting of 69 text characters or less. Select "C" to save the file.

To decrypt the file, run Top Secret and load the encrypted program from the disk. Enter "`SYS 2049`" then enter the password. Press "D" to decrypt the program. Type "`RUN`" to run the program.

Tracks and Sectors

Note: Track 18 is where the directory information is stored.

Track#	Sector#
1-17	0-20
18-24	0-18
25-30	0-17
31-35	0-16

Unscratch

If you accidentally delete a file you can recover it with Disk Rescue from https://commodore.software. Disk Magic can also unscratch files.

Using the 1541 Disk Drive

The default drive in VICE 3.3 is is the CBM 1541 and uses .D64 files as the emulated disk.

1) To list the drive directory, enter the following commands:

```
LOAD "$",8
LIST
```

2) To save a program to the disk:

```
SAVE "NAME",8
```

3) To load a program from the disk:

```
LOAD "NAME",8
```

4) To delete (scratch) a file from the disk:

```
OPEN 1,8,15,"S:FILENAME":CLOSE1
```

Note: you may use the wildcard “*” symbol in the filename to delete multiple files at once.

5) To rename a file on the disk:

```
OPEN 1,8,15,”R:NEWNAME=OLDNAME”:CLOSE1
```

6) To save & replace a file (overwrite):

```
SAVE”@0:NAME”,8
```

7) To format and name a disk:

```
OPEN 1,8,15,”N:DISK_NAME,01”:CLOSE1
```

8) To load and auto run a program:

```
LOAD “filename”,8,1 or LOAD “filename”,8:
Instead of hitting RETURN, press SHIFT + RUN/STOP
```

Wildcard Directory Listings

You are able to customize the output of the LOAD directory command using wildcards. Wildcards allow you to LOAD filenames that match a pattern or show only PRG, SEQ or USR files:

```
LOAD”$0:PROG*”,8

LOAD”$*=P”,8

LOAD”$*=S”,8

LOAD”$*=U”,8
```

Eighty Column Screen

64 Eighty

64 Eighty is a program to enable an 80 column screen on the Commodore 64 published in Compute! Magazine, August 1987. It starts with a black/blue screen but the colors can be adjusted in the normal way.

Left Arrow Functions

```
{Left Arrow}+A Enable auto-insert mode
{Left Arrow}+C Disable auto-insert mode
{Left Arrow}+D Delete line at cursor location
{Left Arrow}+I Insert line at cursor location
{Left Arrow}+E Stop cursor flashing
{Left Arrow}+F Turn on cursor flashing
{Left Arrow}+@ Clear screen from cursor to bottom
{Left Arrow}+Q Erase line from cursor to end of line
{Left Arrow}+P Erase line from cursor to start of line
{Left Arrow}+G Enable bell sound with CTRL+G
{Left Arrow}+H Disable bell sound with CTRL+G
{Left Arrow}+J Move cursor to start of line
{Left Arrow}+K Move cursor to end of line
{Left Arrow}+L Enable screen scrolling
{Left Arrow}+M Disable screen scrolling
{Left Arrow}+O Cancel:quote,insert,underline,reverse,bold
{Left Arrow}+U Change cursor to underline
{Left Arrow}+S Change cursor to block
{Left Arrow}+V Scroll screen up
{Left Arrow}+W Scroll screen down
```

80 Columns 19XX

Another 80 column program available at archive.org:

https://archive.org/details/d64_80_Columns_19xx_-

Screen-80

This program was published in Compute's Gazette, September 1984, Issue 15. When 80 column mode is activated, the block cursor becomes an underline. Custom-80 is a program from the same issue that allows you to create you own 80 column font for use with Screen-80.

`CTRL+N`	Enter lower/uppercase mode
`POKE646,X`	Change text color
`POKE53280,X`	Change border color
`POKE53281,X`	Change background color
`RUN/STOP+RES`	Exit 80 column mode
`SYS710`	Start Screen-80

Fonts

64 Character Set

Your Commodore Magazine Issue 10, July 1985, provided an extra character set to replace some of the graphics characters. Just press the shift key to access this new character set.

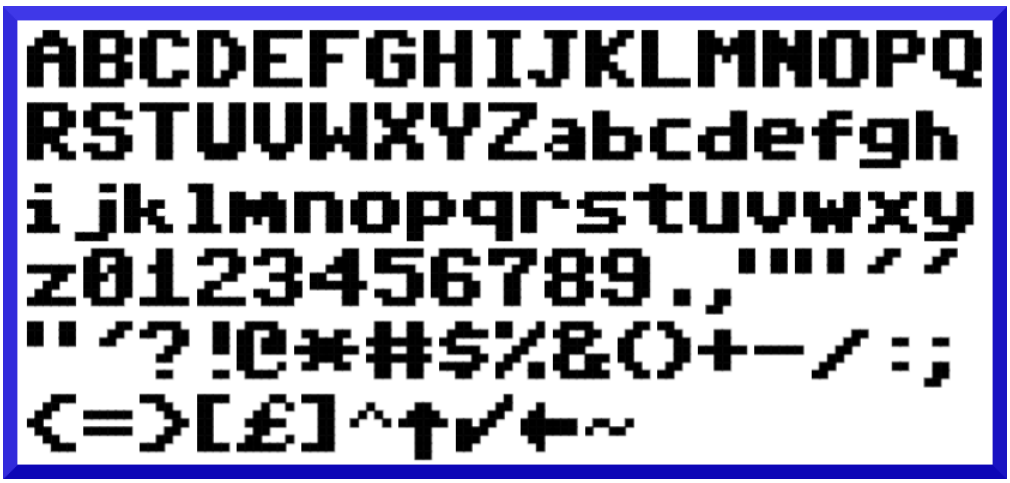
Mini Font Editor

This program was published in RUN magazine as $4B8 and allows you to create custom characters inside a basic program.

MyC64

MYC64 is a ttf font I created mimicking the C64 and is compatible with Windows and Linux PCs. The British pound symbol is the backslash, the curly brackets are the Commodore arrows and the pipe (vertical line above the backslash) is the Commodore check mark. It can be downloaded at:

https://archive.org/details/my-commodore-64-font



UltraFont+

Use UltraFont+ from Compute's Gazette, September 1986, to create custom C64 fonts for use in basic programs or SpeedScript. The cursor keys are used to select the character you want to edit. A joystick in port 2 is used to select a pixel and the fire button is used to turn the pixel on or off.

When you are ready to save the character set just tap the "S" key. Pressing "CTRL+D" will create data statements at the end of any basic program in memory. Just press "CTRL+X" to exit UltraFont+ then type "LIST" to see the data statements. They will begin at line number 63000. There will only be data statements for the characters you modified.

https://youtu.be/RW53m1B9wG8

Games

1000 Miler

This is a computer game based on the classic card game Mille Bornes, simulating a car race. It is available on LoadStar Disk 37.

Bard's Tale I: Tales of the Unknown

A D&D type game with graphics. The old Apple version is available in the Google Play store and the original C64 version along with "Bard's Tale Character Editor" can be downloaded at archive.org.

https://youtu.be/Fs35kuVFpyk

Vampire lord
The sudden scream of battle brings your party to a halt. You face 1 Vampire lord.
Will your stalwart band choose to Fight or Run?
Character Name AC Hits Cnd SpPt CL
BRIAN THE FIST LO 599 599 0 Pa
SAMSON LO 545 545 0 Wa
ROB ZOMBIE LO 679 679 0 Ba
SAM HAIN LO 507 507 422 Ma
JOE THE BRUCE LO 476 476 420
LORD MHORAM LO 350 350 337

Crazy Eights

A cool card game I played as kid. It's available at archive.org.

Cribbage Master

Published in Commodore Disk User, November 1988.

Godzilla

Godzilla is a game where you have to defend Tokyo from total destruction. You can use troops, planes and ships to attack Godzilla. You can use an atom bomb as a last resort, but it doesn't always kill him. This is a type in program from a book and is available for download from archive.org.

Logic

Mastermind clone from Commodore Disk User on disk CDU2-02A.

Mancala

There is a decent Mancala game called “Kalah” in the March/April 1989 Rerun disk issued by RUN Magazine. “Wari” is another Mancala clone and is more colorful than Kalah. Both are available at Archive.org.

Silent Service

Command a submarine during WWII.

Spy Vs. Spy

A fun game based on the comic strip from Mad Magazine. A tip for the one player game is to booby trap all the doors in the room with the door to the airport. When the other spy has been killed, pick up the briefcase and head for the airport.

Yahtzee

There is a cool Yahtzee game called “Yatzy” on the March/April 1989 Rerun disk issued by RUN Magazine. It’s available at Archive.org.

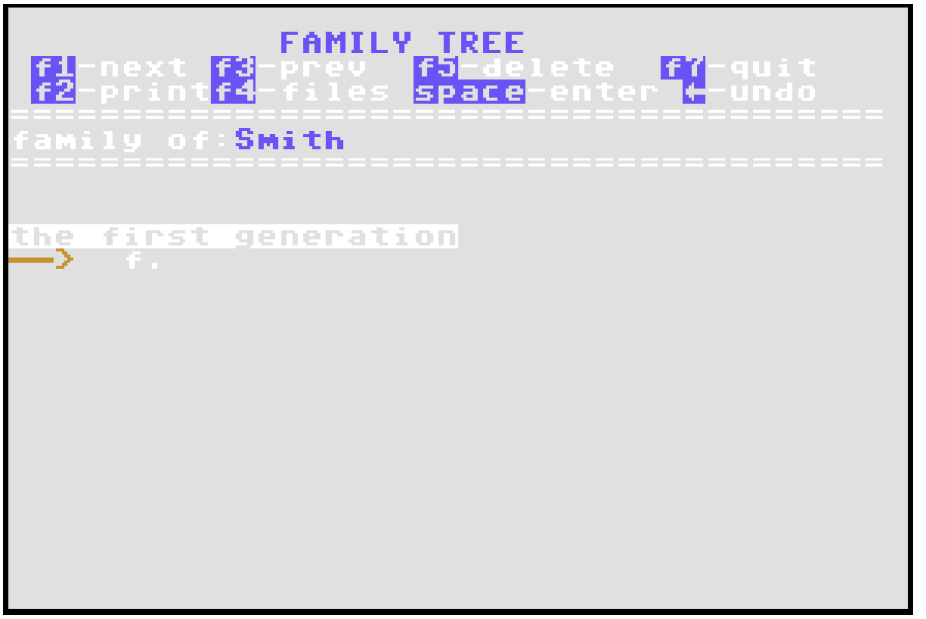
Zaxxon

Type RED at the title screen to be indestructible.

Misc Programs

Family Tree

The Rerun disk from November/December 1989 has a program called “Family Tree” that can hold 400 names and 22 generations.



RUN Paint

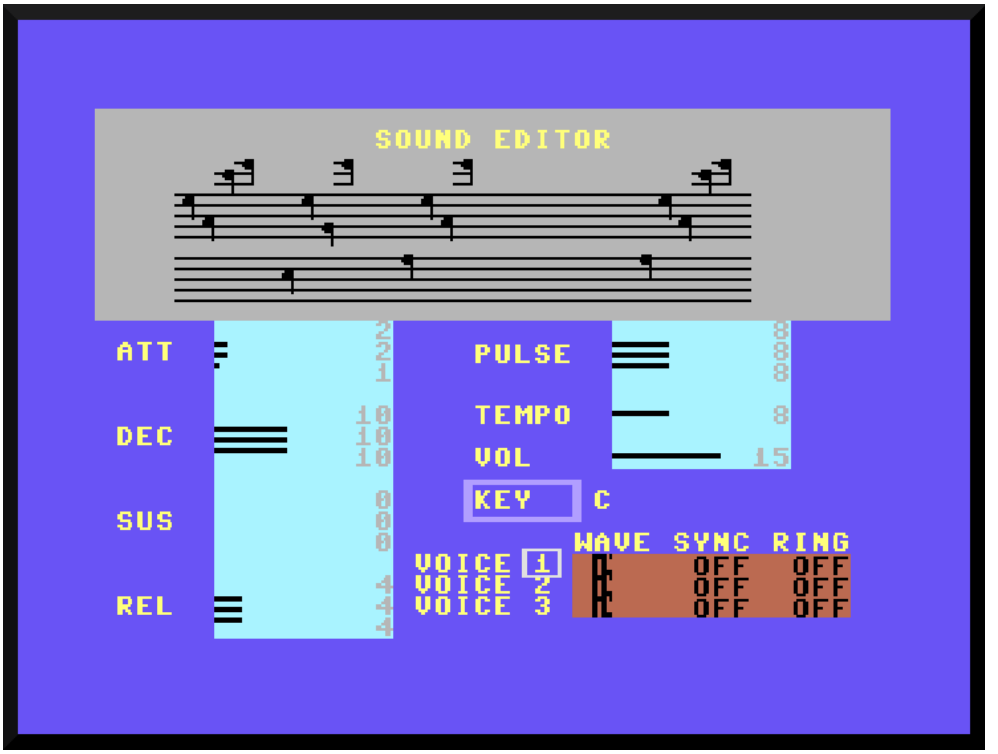
RUN Paint is a joystick based paint program from the March 1989 issue of RUN Magazine. It uses pop-down down menus and uses hi-res and multi-color screens. To use custom fonts, rename the font name to begin with “RPF.” and select it through the options menu. To access the menus, move the arrow up and off the screen and press the fire button.

Music

Basically Music

Adds music commands to C64 basic and contains a sound editing screen. Published in Compute's Gazette Issue 57, March 1988. After the program loads, type RUN and you can load the demo or enter your music commands. The demo is called "FUR ELISE.DEMO". To show the editing screen you must include the SCREEN command at the beginning of your basic program. Keyboard commands:

F1 Move parameter pointer
F3 Move voice pointer
F5 Raise value of selected parameter
F7 Lower value of selected parameter
V Change voice
R Toggle ring modulation
S Toggle synchronization



Music Construction Set

Music Construction Set allows you to write your own music and the C64 will play it for you. To get started just drag and drop the notes on the staffs. Click the piano icon to play your composition.

Music Keyboard

Use the on-screen controls to customize attack, decay, filters, octaves, pulse rate, voices, etc. Published in Commodore User Magazine Issue 33, June 1986.

Music Machine

A cartridge that turns your keyboard into a music keyboard. You can add a background beat or and change the decay, effect, waveform, voices and octave.

Metronome

Hint $448 from RUN magazine turns your C64 into a metronome with a graphics display. You can configure the beats per minute in a range of 40-208.

CRYPT-NOTER V1.0 BY NEW DESIGN/CRYPT
E - EDIT NOTE
D - DIRECTORY
L - LOAD TEXT
S - SAVE TEXT
G - LOAD GRAFIC
C - LOAD CHARSET
M - LOAD MUSIC
F - LOAD SPRITES
P - PACK & SAVE
Z - CLEAR PAGES
W - WORDWARP [0]
I - INFOPAGE

Notepads

Crypt Noter

Crypt Noter is unique in that it allows you to have flashing text in your notes. It is available at archive.org. Commands:

```
SHIFT +/-       Page Up/Down
CBM + I/D       Insert/Delete Line
CBM + L/C/R     Line Left/Center/Right
F1              Normal Character Settings
F3              Flashing 1,2,3
```

Final Cartridge III

Final Cartridge III has a notepad app with pull down menus. It allows saving and loading documents from the disk. Notepad also contains word wrap and line spacing options.

Notemaker

This notepad is from Ahoy! Magazine, April 1986. It makes bootable notes that don't require the original program to be read. They make great "readme.txt" files for your Commodore disks. Use SHIFT+RETURN to advance lines because RETURN will save the note and end your writing. Filenames will have a ".N" attached to the end of their name and must be booted with a ",8,1".

Notepad 64

Notepad 64 allows a full screen per note, 1000 characters. There are options to change the border, background and text colors. All cursor control keys are available. To return to the Notepad 64 menu, use the back arrow key (END on VICE keyboard).

POKEing Around

POKE 1,0	- Disable the system (should be 1,1)
POKE 19,0	- Turn on “?” during INPUT
POKE 19,65	- Turn off “?” during INPUT
POKE 22,35	- LIST without line numbers
POKE 134,1	- Causes errors
POKE 140,0	- Resets the drive
PEEK(152)	- Number of open files
PEEK(186)	- Last used device number
PEEK(197)	- Returns value of key pressed
POKE 198,0	- Clear keyboard buffer
POKE 198,23	- Freezes cursor
POKE 199,0	- Reversed characters off
POKE 199,18	- Reversed characters on
POKE 204,0	- Turn on cursor for GET
POKE 204,255	- Turn off cursor for GET
PEEK(211)	- Get cursor column position
POKE 211,(0-39)	- Set cursor column position
POKE 212,0	- Disable quote mode
PEEK(214)	- Get cursor row position
POKE 214,(0-24)	- Set cursor row position
POKE 646,(0-15)	- Set cursor color
POKE 649,0	- Disables keyboard
POKE 649,10	- Enables keyboard
POKE 650,0	- Restores repeat to cursor keys
POKE 650,64	- Disables repeating keys
POKE 650,128	- Turn on repeating keys
POKE 650,255	- All keys repeat
POKE 651,(0-255)	- Sets delay before key repeat

PEEK(653) - Checks for SHIFT/CBM/CTRL key
- 0 = None
- 1 = SHIFT
- 2 = CBM
- 3 = CBM and SHIFT
- 4 = CTRL
- 5 = CTRL and SHIFT
- 6 = CTRL and CBM
- 7 = CTRL, CBM and SHIFT

POKE 657,00 - Disables SHIFT/CBM
POKE 657,128 - Enables SHIFT/CBM
POKE 770,106:POKE 771,223 - Enable Epyx Fast Load cartridge
POKE 770,131:POKE 771,164 - Disable Epyx Fast Load cartridge
POKE 774,0 - No line numbers with LIST command
POKE 774,64 - LIST protection
POKE 774,134:POKE 775,227 - Disable LIST command
POKE 774,228:POKE 775,252 - Executes cold start with LIST
POKE 774,255 - LIST protection
POKE 775,167 - Enable LIST command
POKE 775,168 - Disable LIST command
POKE 775,200 - Executes CLR/HOME with LIST
POKE 788,49 - Enables RUN/STOP key
POKE 788,51 - Make cursor invisible
POKE 788,52 - Disables RUN/STOP key
POKE 788,80 - Enables drawing with cursor
POKE 801,0:POKE802,0:POKE818,165 - Disable Save
POKE 808,211 - Always return ‘SYNTAX ERROR’
POKE 808,225 - Disables RUN/STOP/RESTORE
POKE 808,234 - Disables RUN/STOP/RESTORE
POKE 808,237 - Enables RUN/STOP/RESTORE
POKE 816,157 - Disables LOAD
POKE 816,165 - Enables LOAD
POKE 818,32 - Disables SAVE command
POKE 818,134:POKE 819,227 - Disables SAVE command
POKE 818,180 - Disables SAVE command

POKE 818,226:POKE 819,252	- Executes cold start with SAVE
POKE 53265,11	- Turn off screen display
POKE 53265,27	- Turn on screen display
POKE 53265,85	- Smaller screen with red cursor
POKE 53272,21	- Enter Uppercase mode
POKE 53272,23	- Enter lowercase mode
POKE 53280,(0-15)	- Set border color
POKE 53280, PEEK(53281)	- Set border color to screen color
POKE 53281,(0-15)	- Set screen color
POKE 53281, PEEK(53280)	- Set screen color to border color
POKE 54296,15:POKE 54296,0	- Make a click sound
POKE 56325,(1-255)	- Cursor speed (Default=50)

List Directory Without Losing Program

Use these commands to load and list the disk directory without losing the BASIC program in memory:

POKE 44,PEEK(46)+1

Now load and list the directory.

POKE 46,PEEK(44)-1:POKE44,8

Your program is now back in memory.

Set Cursor Position

POKE 214, Row#

POKE 211, Column#

SYS58640

Programming Utilities

64 Searcher

Allows you to search a basic program listing for all occurrences of a selected string. It was published Compute's Gazette, September 1983. To use Searcher, enter a line 0 followed be a colon and the text you want to find. Press `RETURN` and enter `SYS49152`. Searcher will then list all line numbers that contain your search string. For example:

```
0: PRINT
```

Abbreviate Zero

If you need to use just the number 0 in a variable or a `POKE` statement, you can abbreviate with a decimal point. The C64 will process the decimal point faster than the 0.

Animated Titles

Tip $357 from RUN magazine contains a short routine to animate text titles in basic programs. There is 26 character limit per line of the title.

Append Two Basic Programs

To append basic programs using `POKE` statements, the second program loaded must have higher line numbers than the first program. Start by loading the first program. When it's loaded, enter:

`PRINT PEEK(43), PEEK(44)` and write down the numbers. Enter:

`X=PEEK(45)+256*PEEK(46)` and also enter the following command:

```
POKE44,INT((X-2)/256):POKE43,(X-2)AND255:NEW
```

Now load the second program and enter the following command:

`POKE43,1:POKE44,8` where 1 & 8 were the two numbers you wrote down in the earlier step. Now you can save the new program.

Append Two Basic Programs With Speedscript

To append to basic programs using Speedscript, the second program must have higher line numbers than the first. Load the first program and move the cursor to the end. You won't see the actual program line entries. All you will see is a screen full of random characters. Delete the last two "@@" symbols (end of file markers) and leave the cursor in place. Load the second program then save your new program to the disk.

Automatic Centering

You can center a text line by entering your message in a variable (V$) and using the following code:

```
10 FOR X=1 TO (40-LEN(V$))/2:PRINT" ";:NEXTX:PRINTV$
```

Change Ready Prompt

To change the ready prompt, type in the program below and run it. Next, enter `POKE1,54`. You can use whatever characters you want in the data statement as long as there are six of them.

```
10 FORX=40960TO49151:POKEX,PEEK(X)
20 NEXTX
30 FORX=41848TO41853:READCH$
40 POKEX,ASC(CH$):NEXTX
50 DATA L,I,N,U,X,!
```

Character Fade & Restore

This little program published in RUN magazine as tip $4D3 makes screen images fizzle in and out.

Clear Returns & Loops

To fix open GOSUB returns and FOR-NEXT loops enter:

```
POKE38,104:POKE39,104:POKE40,76:POKE41,126:POKE42,166:SYS38
```

Cold Start

A could start is the equivalent of turning the C64 off then on. A warm start resets the system pointers.

Color Lister

This program was published in Compute's Gazette, March 1998, and allows you to change the color of the program listing at any point and as many times as you want.

Colorful Listings

To make your program listings have different colors for different parts of your program, insert a special REM statement before the line number where you want the color change to start:

```
REM "”{DEL}{CTRL+9}{SHIFT+M}{COLOR}
```

Use the following chart for the COLOR option above:

```
BLACK          {SHIFT+P}
WHITE          {E}
RED            {#}
CYAN           {CBM+*}
PURPLE         {CBM+-}
GREEN          {UP ARROW}
BLUE           {LEFT ARROW}
YELLOW         {SHIFT+UP ARROW}
ORANGE         {SHIFT+A}
BROWN          {SHIFT+U}
LIGHT RED      {SHIFT+V}
DARK GREY      {SHIFT+W}
MED GREY       {SHIFT+X}
LT GREEN       {SHIFT+Y}
LT BLUE        {SHIFT+Z}
LT GREY        {SHIFT++}
```

Copyright Message

These commands will add a message to your program listing that can't be deleted. Lines 1-3 read the message at line 9999 and copy it to an illegal line number, 65535. After you run the program, delete these lines then save your program.

```
1 I=PEEK(45)+PEEK(46)*256-4
2 IF PEEK(I)<>0 THEN I=I-1:GOTO2
3 POKE I+3,255:POKE I+4,255:END
9999 REM BY JOSEPH GORDON
```

Double Quote After Input

To insert a quotation mark after the question mark of an input statement type the following code. This will allow the entering of characters that may cause issues.

```
POKE198,1:POKE631,34:INPUT CH$
```

Dual Screen 64

Allows you to create a second 1K screen accessed by the F1 key. Could be useful for a help screen. This was published in RUN magazine as tip $5E2.

Erase Screen Lines

It is possible to delete lines of text from the screen or copy a line of text from one line paste it to another. In the below examples, LN = Line number to be erased, LC = Line to be copied from and LP = Line number to be pasted to.

```
POKE781,LN:SYS59903
POKE781,LP:SYS59888:POKE172,PEEK(60656+LC):POKE780,PEEK(216+LC):SYS59848
```

Find Character POKE Codes

An easy way to find the POKE code of a keyboard characters is to position the cursor in the upper left corner of the screen and type the character. Move the cursor down a line and enter:

```
PRINT PEEK(1024)
```

Free Memory

Use PRINT FRE(0) to display the amount of free memory. The output of this command will be an integer between -32767 and +32767. Any amount larger than 32767 will show as a negative number. To see the true amount of free memory when you get a negative number, type:

```
PRINT 65538+FRE(0)
```

Get Statement Trick

Sometimes `GET` statements get messed up because there were key-presses still in the keyboard buffer. The way around this is to use `POKE 198,0` to empty the buffer before using `GET`.

Hide Lines In LIST

At the end of your line, add a colon and two quotes. Delete the second quote, press `CTRL+9`. Enter a T for every character on the line you want to be invisible when the program is listed.

Highlighted Menu

RUN magazine published a highlighted menu routine in January 1989 as tip $4FD.

Indenting

To add spaces after a line number, insert a shifted character followed by the number of spaces you want. The spaces will now appear in the listing. A second way to do this is to add a colon after the line number, then any number of spaces.

INPUT Default Value

You can print a default value next to your `INPUT` statement by using the below examples:

```
10 INPUT"ENTER NUMBER {2 SPACE}8{3 LEFT}";X
10 INPUT"(Y)ES OR (N)O{2 SPACE}Y{3 LEFT}";X$
```

Inputting Commas

The standard INPUT statement will return an error if a user tries to enter a colon, comma, etc. The following codes will allow these characters:

```
10 POKE 19,1:POKE 631,34:POKE 198,1
20 INPUT”QUESTION”;A$
30 PRINT:POKE 19,0
40 PRINT A$
```

or

```
10 POKE631,34:POKE632,34:POKE633,20:POKE198,3
```

Where “,34” is a quote, “,20” is a delete and “,3” is the number of preceding POKES to the keyboard buffer.

INPUT Without Question Mark

```
10 OPEN1,0
20 PRINT”QUESTION”;:INPUT#1,A$
30 END
or
10 PRINT”QUESTION”;
20 POKE631,34:POKE198,1
30 INPUTA$
```

Joystick

To map a gamepad to the VICE C64 emulator in Linux, you can use QjoyPad. It can store multiple profiles which come in handy since some games only use the keyboard and you can map joystick movements to different keys for different games.

Keyboard Chart:

Movement	Joystick 1	Joystick 2
Up	1	CTRL+CRSR-R
Down	Back Arrow	CTRL+A
Left	CTRL	CTRL+D
Right	2	CTRL+6
Fire	SPACE	CTRL+J

Line Freeze

This little program was published in RUN magazine as tip $590. It allows you to freeze the top six screen lines so they won't be erased.

List Protection

To stop someone from being able to view the listing of your basic program, type '`POKE 2050,0`' before saving it to disk. When the program is loaded, it will run normally but won't list until a '`POKE 2050,8`' is entered. Another way to disrupt listing programs is to insert a `REM SHIFT-L` statement, or try:

Add these two lines to the beginning of the BASIC program:

```
0 SYS2063
1 **********
```

Then enter the following POKES in direct mode:

```
POKE 2063,169:POKE 2064,26:POKE 2065,133:POKE 2066,43
POKE 2067,32:POKE 2068,89:POKE 2069,166:POKE 2070,76
POKE 2071,174:POKE 2072,167:POKE2059,0:POKE 2060,0
```

Load Disk Menu From Program

This is a neat way to automatically go back to the disk menu when exiting a basic program. Instead of ending a program with an `END` statement, use the program below to load the disk menu:

```
10 POKE646,PEEK(53281):PRINT"{CLR}{DN}{DN}{DN}NEW:{DN}{DN}"
20 PRINT"LOAD"+CHR$(34)+"MENU64"+CHR$(34)+",8,1"
30 PRINT"{DN}{DN}{DN}{DN}{DN}RUN:{HOME}
40 FORX=0TO3:POKE631+X,13:NEXT:POKE198,4
50 END
```

MetaBASIC

MetaBASIC, Compute's Gazette April 1985, adds the following commands to C64 BASIC:

`AUTO start number, increment`

AUTO is an automatic line numbering routine where you enter the starting line number and increment number. The C64 will now automatically enter line numbers for you.

`CAT`

The CATalog command allows you to view the disk directory without wiping out any basic program you may have in memory.

`CHANGE @OLD@NEW@,start,end`

The change command is a replacement command to change the old string in your basic program to a new string between two line numbers.

`DEFAULT Border,Background,Text,Device`

This command allows you to change the border color, background color, text color and default device.

`DELETE Start-End`

This command will delete line numbers from the start number through the end number.

`DLIST "filename"`

DLIST allows you load a program from the disk and list it on the screen without erasing the program you have in memory.

`DUMP`

DUMP shows the current values of all non-array variables. Use `CONT` to resume execution of the program.

`ERR`

ERR displays the contents of the disk error channel.

`FIND @string@,start,end`

Find will search between the starting and ending line numbers for a string and display all matching lines on the screen.

`HELP`

Displays a list of MetaBASIC commands.

`KEY`

The KEY command allows you to program the function keys. For example, `KEY1,"LIST"` will print LIST on the screen whenever the F1 key is pressed.

`LLIST`

This command will send the program listing to the printer.

`READ "filename"`

The READ command will list a sequential file on the screen without erasing the basic command in memory.

`RENUM start,increment`

This command will renumber a basic program with a new starting line number counting by the increment number.

`RESAVE "filename"`

The RESAVE command deletes the old file from the disk and saves the updated file from memory.

`SCRATCH "filename"`

Deletes a file from the disk.

`START "filename"`

Displays the starting address for a file on the disk.

`TRACE`

Displays the line number currently being executed.

`UNNEW`

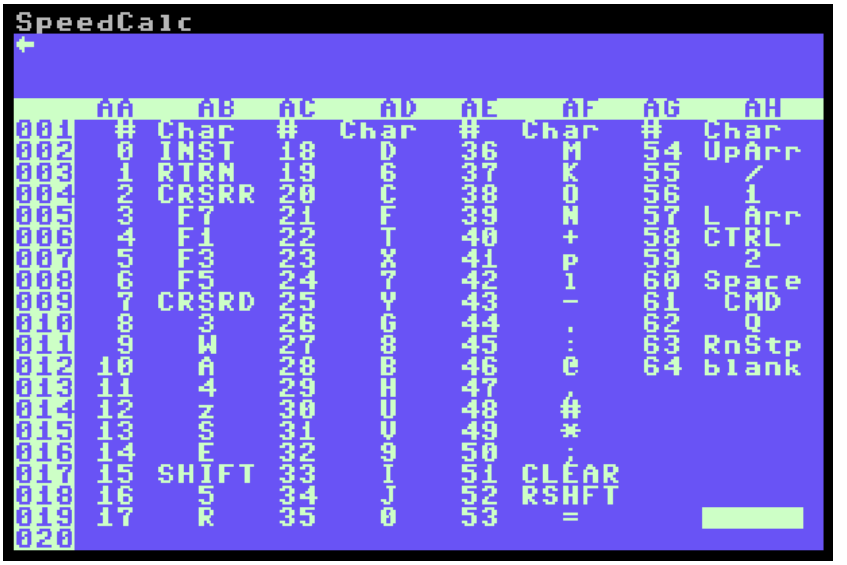
Restores a basic program accidentally erased by a `NEW` command.

`VCHANGE`

Same as the `CHANGE` command except you must approve each and every change with Y or N.

PEEK(197) Table

`PEEK(197)` is used to scan the keyboard to determine what key is being pressed. 197 holds the key pressed before the current key which is `PEEK(203)`. The `SHIFT`, `CTRL` and `CMD` keys use address 653 for the current key press and 654 for the previous key press. The table below shows the number of the PEEK value and its corresponding character.



	AA	AB	AC	AD	AE	AF	AG	AH
001	#	Char	#	Char	#	Char	#	Char
002	0	INST	18	D	36	M	54	UpArr
003	1	RTRN	19	6	37	K	55	/
004	2	CRSRR	20	C	38	O	56	1
005	3	F7	21	F	39	N	57	L Arr
006	4	F1	22	T	40	+	58	CTRL
007	5	F3	23	X	41	P	59	2
008	6	F5	24	7	42	1	60	Space
009	7	CRSRD	25	Y	43	-	61	CMD
010	8	3	26	G	44	.	62	Q
011	9	W	27	8	45	:	63	RnStp
012	10	A	28	B	46	@	64	blank
013	11	4	29	H	47	,		
014	12	Z	30	U	48	#		
015	13	S	31	V	49	*		
016	14	E	32	9	50	;		
017	15	SHIFT	33	I	51	CLEAR		
018	16	5	34	J	52	RSHFT		
019	17	R	35	0	53	=		
020								

PRINT@

To start printing at a specific row and column number, use the following code where x is the row number and y is the column number:

```
POKE781,X:POKE782,Y:POKE783,0:SYS65520:PRINT”TEXT”
```

or

```
POKE214,X:POKE211,Y:SYS58732:PRINT”TEXT MESSAGE”
```

Programming Function Keys

Your Commodore Magazine Issue 4, January 1985, includes a small type-in program to assign strings to all eight function keys. Just put them in the data statements at the end of the program. Examples are provided.

You can also try “Easy Keys” from Commodore Magazine Issue 15, March 1988 or “Functions” from Commodore Disk User, Issue 22. Functions allows to program 16 function keys.

“Func*Keys” is another function key program tool that lets you save your key settings to disk. It was published by Commander Magazine Issue 18, June 1984.

Function Key Magician from Compute’s Gazette Issue 56, February 1988, also allows the programming of 16 keys. It also allows you to save your settings to disk. Type `SYS51200` to start the program.

Quote Mode

To get of quote mode, use `SHIFT+RETURN` then use the cursor to go back to the line you were working on or use `SHIFT+2` then `DELETE`.

Random Numbers

The Commodore 64 is capable of generating random numbers to simulate dice rolls or anything else that you need them for. The below example will produce a random number between one and six:

```
RN=INT(RND(1)*6)+1
```

REDIM'D ARRAY ERROR

The array pointer must be reset before DIMs can be re-dimensioned. The following command will reset the array pointer without clearing the other variables:

```
POKE49,PEEK(47):POKE50,PEEK(48)
```

Recover After NEW Command

If you accidentally wipe out the program in memory with the `NEW` command before saving it, it's not lost forever. Just use the following commands:

```
POKE2050,1:SYS42291:POKE45,PEEK(34):POKE46,PEEK(35):CLR
```

Renumber

The Renumber program from Loadstar, Issue 11, Side 1, PowerPlay Programs – May 1985 will renumber an entire Basic program. It will ask for starting and increment numbers and take care of the rest, including `GOTO` and `GOSUB` number jumps.

Screen Memory Location Finder

I wrote a short program to help correctly identify the screen and color memory map codes. Just enter the column number and row number into the calculator and you will receive the correct location codes to poke character and color to the correct screen locations.

```
5 REM SCREEN MEMORY LOCATION FINDER
10 PRINT"{CLR/HOME}{CYAN}"
20 PRINT" "
30 INPUT"COLUMN NUMBER:";C
35 IF C<0 OR C>39 THEN 30
40 INPUT"ROW NUMBER:";R
45 IF R<0 OR R>24 THEN 40
50 SM=1024+C+(40*R):CM=55296+C+(40*R)
60 PRINT" "
62 PRINT"SCREEN LOCATION:";SM
63 PRINT"COLOR LOCATION: ";CM
```

```
70 PRINT" "
80 PRINT"ANOTHER? (Y/N)"
85 GETA$:IFA$="" THEN 85
90 IFA$="Y" THEN 10
100 PRINT"{CLR/HOME}":END
```

Screen Scroll

To move the screen up one line use: `SYS59626`

Screen Shake

A small routine to shake the screen:

```
10 FOR X=0TO255:POKE53720,X:NEXTX:POKE53720,200:END
```

Sequential File Editor

The Sequential File Editor from Compute's Gazette Special 1988a disk will allow you to view and edit a sequential file. It also allows listing with and without line numbers.

Show Directory Without Losing Program

To show the directory without losing the program currently in memory:

```
POKE 44,PEEK(46)+1
```

Now you can load the directory and list it. When you want to return your program to memory type:

```
POKE 46,PEEK(44)-1:POKE44,8
```

Split 64

Split 64 from Your Commodore Magazine Issue 14, November 1985, has a type-in program to divide the basic area into to separate 16K computers. Use `SHIFT+CTRL` to switch between work spaces. `RUN/STOP+RESTORE` will end the program. Type `SYS 35896` to resume Split 64 or `SYS 35840` to reset.

Start/End Address

To find the start address of a basic program in memory:

```
PRINT PEEK(43)+PEEK(44)*256
```

To find the end address of a basic program in memory:

```
PRINT PEEK(45)+PEEK(46)*256
```

To find the high and low bytes of an address, use the “Hi Lo Byte Finder” program I wrote: https://archive.org/details/hi-low-byte-finder

Text Centering

Here is a small routine to center text on the screen:

```
10 INPUT ”YOUR TEXT”;X$
20 X=LEN(X$)
30 PRINTTAB((40-X)/2)X$
40 END
```

Time Variable

TI$ is where the C64 saves the time information. It is six digits and uses military time (24 hour clock). To set the time, type:

```
TI$=″HHMMSS″
```

Trace

To help debug your basic program, just add "`SYS 49578`" to various points in your program. When the line is executed it will print to the screen "`IN`" followed by the line number just executed.

Triple 64

Triple 64 was published in Compute's Gazette, April 1985. It creates three separate 12K computers allowing a programmer to work on three basic programs at once. To switch between these virtual machines just type `SYS 40004` and enter 1, 2 or 3.

Universal Input

Universal Input is a machine language routine published in Compute's Gazette Issue 77, November 1989. It solves the problem of entering special characters in INPUT statements. It also increases the maximum number of characters allowed from 80 to 255.

Spreadsheets

CalcAid 64

CalcAid 64 is a spreadsheet program published in RUN magazine, November 1986. It has 30 columns x 26 rows. The maximum number of characters per cell is nine. If using a Commodore 128 emulator, see the March 1987 edition of RUN magazine, issue 39, pg. 82, and Hint $3C3,for updated code that enables you to view all seven columns on the 80 column screen. Use “GRAPHIC5” to switch to 80 columns, then load and run CalcAid.

To enter text type the cell number, a colon and the text (A1:Text). To enter numbers type the cell number, a colon and the number (A2:128). Text data cannot begin with a number or +/-, and numerical data cannot begin with a text character. The format for formulas is cell number, colon, F1, formula (A10:{F1}A1+A2).

```
F2 – Rounds numbers
F3 – Load
F4 – Save
F5 – Print
F6 – Print formulas
F7 – Background color
F8 – Border color
CTRL+(1-8) – Text color
Home – Move cursor to cell A0
SHIFT+CLR – Erase sheet
A0:{F1}T – Makes cell A0 a title
A0:{F1}O – Turns off title mode
A1:{F1}C – Clear a cell A1
A2:{F1}COPA5-A10 – Copies cell A2 to cells A5 thru A10
A11:{F1}V – View formula in cell A11
Left Arrow – Recalculate sheet starting from column A
SUM – Sum a column of cells
AVG – Calculate the average of cells
MIN – Calculate minimum value
MAX – Calculate maximum value
```

SpeedCalc

SpeedCalc is a nice spreadsheet for the C64 with the capability of exporting cell blocks to a SpeedScript document. It has 50 columns and 200 rows. It's free software from Compute's Gazette disk, March 1994.

Export Cell Blocks

To export from and SpeedCalc and import to SpeedScript requires a little effort. Put the cursor directly below and to the right of the cell blocks you want to export and hit SHIFT+CTRL+P and choose "D" to print the cells to the disk. Next, run SpeedScript Integrator to convert the cells you just printed to SpeedScript format. Lastly, run SpeedScript and load the integrated file.

Keyboard Controls

```
F1-Border colors
F3-Background colors
F5-Text colors
F7-Load
F8-Save
CTRL+4 - Display directory
CTRL+A - display free memory
CRTL+B - Erase current cell
CTRL+C - Copy
CTRL+E - Edit cell contents
CTRL+F - Change format (Left, Center or Right) & # of decimal places
CTRL+G - go to cell 'xxxxx'
CTRL+M - Move
CTRL+P - Print sheet (Cursor below and right of block you wish to print)
CTRL+R - Turn recalculation on
CTRL+T - Change cell type (Text, Numeric or Formula)
CTRL+W - Change width of a column (4-36)
CTRL+X - Exit SpeedCalc
CTRL+Up Arrow - Send disk command
SHIFT+CLR/HOME - Erase sheet
SHIFT+CTRL+C - Relative copy, adjusts cell names in formulas
SHIFT+CTRL+F - Global format
SHIFT+CTRL+M - Relative move, adjusts cell names in formulas
SHIFT+CTRL+P - Preview spreadsheet or print to disk for Speedscript
SHIFT+CTRL+R - Displays recalculation mode
SHIFT+CTRL+W - Global width
Left Arrow - Recalculate sheet
```

Functions

```
+ Addition
- Subraction
* Multiplication
/ Division
Up Arrow - Exponent
= Equality
@abs()        Absolute value
@atn()        Arctangent
@ave()        Average of a block of cells
@cos()        Cosine
@exp()        Value of e (2.7182318...)
@int()        Integer
@log()        Natural logarithm
@sgn()        Sign
@sin()        Sine
@sqr()        Square root
@sum()        Sum block of cells =@sum(aa1:aa20) adds aa1 thru aa20
@tan()        Tangent
pi            Value of pi (3.14159265)
```

Notes:

- Use lower case letters when entering cell names, ex 'aa1'

- Press HOME key to go to upper left cell, press twice to go to cell aa1

- If using VICE, you may have to use the positional keyboard

- No commas with numbers and no scientific notation

- The first character of a formula must be "="

- ***** Number is too large to display in the cell

- For Copy and Move: Move cursor to the upper left cell of the block you want to copy or move and press CTRL+C or CTRL+M. Command lines turns purple. Move cursor to the lower right cell of the block to copy or move and press RETURN. Move the cursor to the upper left corner of where you want to paste to and press RETURN.

SwiftCalc 128

- F9 is the “Escape” key.

SYSops

SYS 42115	- End program without the READY prompt
SYS 44808	- Causes a Syntax Error
SYS 57194	- Enables Epyx Fast Load cartridge
SYS 58235	- Performs warm start
SYS 58260	- Resets system
SYS 58726	- CLR/HOME
SYS 59062	- Advances cursor
SYS 59137	- Go to previous line
SYS 59626	- Move all text up one line
SYS 59903	- Clear line
SYS 62913	- Name of last file loaded
SYS 64738	- Resets pointers, colors & random numbers
SYS 65126	- Alternate start
SYS 65499	- Sets TI$ to zero
SYS 65511	- Closes all files

WAIT for it

SLEEP 0	- Wait for next interrupt
SLEEP (1-65535)	- Halt program for X amount of seconds
WAIT 1,32,32	- Wait until a datassette button is pressed
WAIT 145,1,1	- Wait for joystick 1 UP
WAIT 145,2,2	- Wait for joystick 1 DOWN
WAIT 145,4,4	- Wait for joystick 1 LEFT
WAIT 145,8,8	- Wait for joystick 1 RIGHT
WAIT 145,16,16	- Wait for joystick 1 FIRE
WAIT 145,31,31	- Wait for any joystick 1 action
WAIT 198,1	- Wait until a key is pressed
WAIT 513,255,PEEK(513)	- Waits for timer to advance 4.2 seconds
WAIT 515,255,PEEK(515)	- Waits for any key
WAIT 516,1,PEEK(516)	- Waits for SHIFT key
WAIT 516,255	- Wait until a key is pressed
WAIT 525,1:POKE 525,0	- Wait until a key is pressed
WAIT 525,2:POKE 525,0	- Wait for two keys to be pressed
WAIT 653,1	- Wait until the SHIFT key is pressed
WAIT 653,2	- Wait until the Commodore key is pressed
WAIT 653,3	- Wait until the CTRL key is pressed
WAIT 53273,6,6	- Wait for sprite collision with characters
WAIT 56464,1,1	- Wait for joystick 2 UP
WAIT 56464,2,2	- Wait for joystick 2 DOWN
WAIT 56464,4,4	- Wait for joystick 2 LEFT
WAIT 56464,8,8	- Wait for joystick 2 RIGHT
WAIT 56464,16,16	- Wait for joystick 2 FIRE
WAIT 56464,31,31	- Wait for any joystick 2 action
WAIT 59410,255,251	- Wait until the spacebar is pressed
WAIT 59411,8,8	- Wait for play on datassette
WAIT 59411,8	- Halt while a datassette button is down

Windows

64 Windows

Allows the programmer to create a pop-up window starting at the top of the screen. You can program the column for the left and right side of the box and the line number for the bottom. Colors are also configurable. Use SYS49152 to copy the screen, SYS49263 to activate the window and SYS49335 to restore the screen. This program is available in the January 1989 edition of RUN magazine.

Pop-Up Menus

Another windowing program from Commodore Disk User, September 1991, Issue 35. It allows the programmer to design pop-up windows in basic programs.

Quick Windows

Published on the ReRun Summer 1990 disk, Quick Windows allows pop-up windows on the Commodore 64. Some options are a solid box, hollow box, extended box with different colors for border and interior, and a plot at routine to position the cursor. It also contains a memory mover to store screens.

Window Wizard

Window wizard has 19 commands to help a programmer create pop-up windows in C64 basic programs. It was published In Compute's Gazette, September 1986. It's a little complicated but works really well and is very customizable.

Enigmuh V2.1 Program Listing

```
https://archive.org/details/@josephgordon
 10 DIMCC$(64):DIMCP(64)
 20 DIM ME$(255):CO=0
 30 REM ENCRYPT UP TO 240 CHARACTERS
 40 REM INCLUDES : / . @ AND SPACE
 50 REM BLUE LIGHT - CAN BEGIN TYPING
 60 REM LIGHT UP TYPED LETTER
 70 REM ENCODER WHEELS ADVANCE
 80 REM BUZZER 4 UNSUPPORTED CHARACTER
 90 REM CLICK FOR ACCEPTED CHARACTER
100 REM TAKE SCREENSHOT OF ENCRYPTED -
110 REM TEXT & POST TO SOCIAL MEDIA
120 REM *** SET VARIABLES
130 CN=0:DIMCH$(64):W1$="":W2$=""
140 W3$="":W1=65:W2=65:W3=65:W4=48
150 W5=48:Y=0:WC=0:GC=0
160 P1$="":P1=0:P2=0:PB=0
170 REM *** READ CHARACTERS
180 FORX=0TO63:READCH$(CN)
190 CN=CN+1:NEXTX
200 REM *** GET WHEEL SETTINGS
210 PRINT"{CLR}{CYN}"
220 PRINT"NOTE: CHARACTER LIMIT IS 240"
230 PRINT"& WILL SHOW '6O' ON COUNTER"
```

```
240 PRINT"F1 TO PRINT CODE, WILL AUTO"
250 PRINT"PRINT WHEN LIMIT IS REACHED"
260 PRINT
270 PRINT"USE 0-9 & A-Z FOR WHEEL CODE"
280 PRINT"ENTER CODE FOR WHEEL 1"
290 GETA$:IFA$=""THEN290
300 GOSUB4810
310 IFA1<1THEN290
320 IFA1>26ANDA1<48THEN290
330 IFA1>57THEN290
340 W1=A1
350 PRINT"ENTER CODE FOR WHEEL 2"
360 GETA$:IFA$=""THEN360
370 GOSUB4810
380 IFA1<1THEN360
390 IFA1>26ANDA1<48THEN360
400 IFA1>57THEN360
410 W2=A1
420 PRINT"ENTER CODE FOR WHEEL 3"
430 GETA$:IFA$=""THEN430
440 GOSUB4810:PRINT""
450 PRINT"(E)NCRYPT OR (D)ECRYPT"
460 PRINT""
470 GETD$:IFD$=""THEN470
480 IFA1<1THEN430
```

```
  490 IFA1>26ANDA1<48THEN430
  500 IFA1>57THEN430
  510 W3=A1
  520 PRINT"":PRINT"1ST PLUG BOARD CONNECTION
(A,B,C,D)"
  530 GETP1$:IFP1$=""THEN530
  540 IFP1$="A"ORP1$="B"ORP1$="C"ORP1$="D"THEN560
  550 GOTO530
  560 PRINT"":PRINT"2ND PLUG BOARD CONNECTION
(1,2,3,4)"
  570 GETP2:IFP2<1ORP2>4THEN570
  580 IFP2<1ORP2>4THEN570
  590 WC=INT(W1+W2-W3):IFWC<=0THENWC=1
  600 PRINT"{CLR}":POKE53265,11
  610 REM *** POKE W1,W2,W3,W4,W5
  620 POKE1186,W1:POKE55458,1
  630 POKE1190,W2:POKE55462,1
  640 POKE1194,W3:POKE55466,1
  650 POKE1198,W4:POKE55470,1
  660 POKE1202,W5:POKE55474,1
  670 REM *** DRAW SCREENS
  680 REM *** DRAW LINES
  690 POKE53281,0:REM BLACK BACKGROUND
  700 POKE53280,12:REM GREY2 BORDER
  710 Y=55296:FORX=1024TO1063:REM LINE1
```

```
720 POKEX,102:POKEY,8:Y=Y+1:NEXTX
730 Y=55616:FORX=1344TO1383:REM LINE2
740 POKEX,102:POKEY,8:Y=Y+1:NEXTX
750 Y=55660:FORX=1388TO1508STEP40
760 POKEX,102:POKEY,8:Y=Y+40:NEXTX
770 Y=55691:FORX=1419TO1539STEP40
780 POKEX,102:POKEY,8:Y=Y+40:NEXTX
790 Y=55816:FORX=1544TO1584:REM LINE3
800 POKEX,102:POKEY,8:Y=Y+1:NEXTX
810 Y=55296:FORX=1024TO1984STEP40
820 POKEX,102:POKEY,8:Y=Y+40:NEXTX
830 Y=55335:FORX=1063TO2023STEP40
840 POKEX,102:POKEY,8:Y=Y+40:NEXTX
850 Y=56256:FORX=1984TO2023
860 POKEX,102:POKEY,8:Y=Y+1:NEXTX
870 REM *** DRAW WHEELS
880 Y=55378:FORX=1106TO1122STEP4
890 POKEX,114:POKEY,15:Y=Y+4:NEXTX
900 Y=55417:FORX=1145TO1163STEP4
910 POKEX,85:POKEY,15:Y=Y+4:NEXTX
920 Y=55418:FORX=1146TO1162STEP4
930 POKEX,113:POKEY,15:Y=Y+4:NEXTX
940 Y=55419:FORX=1147TO1163STEP4
950 POKEX,73:POKEY,15:Y=Y+4:NEXTX
960 Y=55457:FORX=1185TO1203STEP2
```

```
 970 POKEX,66:POKEY,15:Y=Y+2:NEXTX
 980 Y=55497:FORX=1225TO1241STEP4
 990 POKEX,74:POKEY,15:Y=Y+4:NEXTX
1000 Y=55498:FORX=1226TO1242STEP4
1010 POKEX,114:POKEY,15:Y=Y+4:NEXTX
1020 Y=55499:FORX=1227TO1243STEP4
1030 POKEX,75:POKEY,15:Y=Y+4:NEXTX
1040 Y=55538:FORX=1266TO1282STEP4
1050 POKEX,91:POKEY,15:Y=Y+4:NEXTX
1060 REM *** POKE CHARACTERS
1070 PRINT"{HOME}{DOWN}{DOWN}{DOWN}{RGHT}{RGHT}{RGHT}
{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}
{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}
{RGHT}{RGHT}V2.1"
1080 PRINT"{HOME}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}
{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}
{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}
{RGHT}{RGHT}{RGHT}{RGHT}{RGHT}ENIGMUH CIPHER{CYN}"
1090 POKE1395,17:POKE55667,7:REM Q
1100 POKE1397,23:POKE55669,7:REM W
1110 POKE1399,5 :POKE55671,7:REM E
1120 POKE1401,18:POKE55673,7:REM R
1130 POKE1403,20:POKE55675,7:REM T
1140 POKE1405,26:POKE55677,7:REM Z
1150 POKE1407,21:POKE55679,7:REM U
1160 POKE1409,9 :POKE55681,7:REM I
1170 POKE1411,15:POKE55683,7:REM O
```

```
1180 POKE1413,46:POKE55685,7:REM .
1190 POKE1436,1 :POKE55708,7:REM A
1200 POKE1438,19:POKE55710,7:REM S
1210 POKE1440,4 :POKE55712,7:REM D
1220 POKE1442,6 :POKE55714,7:REM F
1230 POKE1444,7 :POKE55716,7:REM G
1240 POKE1446,8 :POKE55718,7:REM H
1250 POKE1448,10:POKE55720,7:REM J
1260 POKE1450,11:POKE55722,7:REM K
1270 POKE1452,47:POKE55724,7:REM /
1280 POKE1475,16:POKE55747,7:REM P
1290 POKE1477,25:POKE55749,7:REM Y
1300 POKE1479,24:POKE55751,7:REM X
1310 POKE1481,3 :POKE55753,7:REM C
1320 POKE1483,22:POKE55755,7:REM V
1330 POKE1484,32:POKE55756,7:REM SPACE
1340 POKE1485,2 :POKE55757,7:REM B
1350 POKE1487,14:POKE55759,7:REM N
1360 POKE1489,13:POKE55761,7:REM M
1370 POKE1491,12:POKE55763,7:REM L
1380 POKE1493,58:POKE55765,7:REM :
1390 POKE1514,48:POKE55786,7:REM 0
1400 POKE1516,49:POKE55788,7:REM 1
1410 POKE1518,50:POKE55790,7:REM 2
1420 POKE1520,51:POKE55792,7:REM 3
```

```
1430 POKE1522,52:POKE55794,7:REM 4
1440 POKE1524,53:POKE55796,7:REM 5
1450 POKE1526,54:POKE55798,7:REM 6
1460 POKE1528,55:POKE55800,7:REM 7
1470 POKE1530,56:POKE55802,7:REM 8
1480 POKE1532,57:POKE55804,7:REM 9
1490 POKE1534,0 :POKE55806,7:REM @
1500 POKE1140,64:POKE55412,12
1510 POKE1139,112:POKE55411,12
1520 POKE1141,110:POKE55413,12
1530 POKE1179,66:POKE55451,12
1540 POKE1181,66:POKE55453,12
1550 POKE1180,81:POKE55452,12:REM G DOT
1560 POKE1219,109:POKE55491,12
1570 POKE1221,125:POKE55493,12
1580 POKE1220,64:POKE55492,12
1590 POKE1136,64:POKE55408,12
1600 POKE1135,112:POKE55407,12
1610 POKE1137,110:POKE55409,12
1620 POKE1175,66:POKE55447,12
1630 POKE1177,66:POKE55449,12
1640 POKE1176,81:POKE55448,12:REM R DOT
1650 POKE1215,109:POKE55487,12
1660 POKE1217,125:POKE55489,12
1670 POKE1216,64:POKE55488,12
```

```
1680 POKE1385,1:POKE55657,12:REM PB-A
1690 POKE1425,2:POKE55697,12:REM PB-B
1700 POKE1465,3:POKE55737,12:REM PB-C
1710 POKE1505,4:POKE55777,12:REM PB-D
1720 POKE1387,49:POKE55659,12:REM PB-1
1730 POKE1427,50:POKE55699,12:REM PB-2
1740 POKE1467,51:POKE55739,12:REM PB-3
1750 POKE1507,52:POKE55779,12:REM PB-4
1760 FORX=55658TO55778STEP40
1770 POKEX,10:NEXTX
1780 IFP1$="A"ANDP2=1THENPOKE1386,64
1790 IFP1$="A"ANDP2=2THENPOKE1386,73
1800 IFP1$="A"ANDP2=2THENPOKE1426,74
1810 IFP1$="A"ANDP2=3THENPOKE1386,73
1820 IFP1$="A"ANDP2=3THENPOKE1426,66
1830 IFP1$="A"ANDP2=3THENPOKE1466,74
1840 IFP1$="A"ANDP2=4THENPOKE1386,73
1850 IFP1$="A"ANDP2=4THENPOKE1426,66
1860 IFP1$="A"ANDP2=4THENPOKE1466,66
1870 IFP1$="A"ANDP2=4THENPOKE1506,74
1880 IFP1$="B"ANDP2=1THENPOKE1386,85
1890 IFP1$="B"ANDP2=1THENPOKE1426,75
1900 IFP1$="B"ANDP2=2THENPOKE1426,64
1910 IFP1$="B"ANDP2=3THENPOKE1426,73
1920 IFP1$="B"ANDP2=3THENPOKE1466,74
```

```
1930 IFP1$="B"ANDP2=4THENPOKE1426,73
1940 IFP1$="B"ANDP2=4THENPOKE1466,66
1950 IFP1$="B"ANDP2=4THENPOKE1506,74
1960 IFP1$="C"ANDP2=1THENPOKE1386,85
1970 IFP1$="C"ANDP2=1THENPOKE1426,66
1980 IFP1$="C"ANDP2=1THENPOKE1466,75
1990 IFP1$="C"ANDP2=2THENPOKE1426,85
2000 IFP1$="C"ANDP2=2THENPOKE1466,75
2010 IFP1$="C"ANDP2=3THENPOKE1466,64
2020 IFP1$="C"ANDP2=4THENPOKE1466,73
2030 IFP1$="C"ANDP2=4THENPOKE1506,74
2040 IFP1$="D"ANDP2=1THENPOKE1386,85
2050 IFP1$="D"ANDP2=1THENPOKE1426,66
2060 IFP1$="D"ANDP2=1THENPOKE1466,66
2070 IFP1$="D"ANDP2=1THENPOKE1506,75
2080 IFP1$="D"ANDP2=2THENPOKE1426,85
2090 IFP1$="D"ANDP2=2THENPOKE1466,66
2100 IFP1$="D"ANDP2=2THENPOKE1506,75
2110 IFP1$="D"ANDP2=3THENPOKE1466,85
2120 IFP1$="D"ANDP2=3THENPOKE1506,75
2130 IFP1$="D"ANDP2=4THENPOKE1506,64
2140 POKE1420,35:POKE55692,7:REM #
2150 POKE1421,36:POKE55693,7:REM $
2160 POKE1422,37:POKE55694,7:REM %
2170 POKE1460,38:POKE55732,7:REM &
```

```
 2180 POKE1461,39:POKE55733,7:REM '
 2190 POKE1462,45:POKE55734,7:REM -
 2200 POKE1500,43:POKE55772,7:REM +
 2210 POKE1501,42:POKE55773,7:REM *
 2220 POKE1502,44:POKE55774,7:REM ,
 2230 POKE1540,63:POKE55812,7:REM ?
 2240 POKE1541,33:POKE55813,7:REM !
 2250 POKE1542,61:POKE55814,7:REM =
 2260 POKE53265,27:REM RESTORE SCREEN
 2270 REM *** INPUT LETTER AND ENCRYPT
 2280 REM *** MAKE CLICK FOR LETTER
 2290 REM *** BUZZER FOR BAD LETTER
 2300 PRINT"{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}
{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}{DOWN}"
 2310 CC=0:WW=0:XL=1665:TW=1
 2320 W4=48:W5=49
 2330 FORX=0TO52
 2340 READCC$(X):NEXTX:REM CHARACTERS
 2350 FORX=0TO52
 2360 READCP(X):NEXTX:REM POKE NUMBERS
 2370 IFD$="D"THEN2750
 2380 POKE1180,81:POKE55452,6
 2390 GETIN$:IFIN$=""THEN2390
 2400 IFIN$=CHR$(133)THENGOTO5360
 2410 FORX=0TO52
```

```
 2420 IFIN$=CC$(X)THENCC=X:GOTO2530
 2430 NEXTX
 2440 IFIN$="<"ORIN$=">"THENGOSUB4480
 2450 IFIN$=CHR$(92)ORIN$=CHR$(94)THENGOSUB4480:REM
POUND UP ARROW
 2460 IFIN$=CHR$(59)ORIN$=CHR$(95)THENGOSUB4480:REM ;
LEFT ARROW
 2470 IFIN$=CHR$(34)ORIN$=CHR$(126)THENGOSUB4480:REM "
AND PI
 2480 IFIN$=CHR$(157)ORIN$=CHR$(29)THENGOSUB4480:REM
LR CURSOR
 2490 IFIN$=CHR$(145)ORIN$=CHR$(17)THENGOSUB4480:REM
UD CURSOR
 2500 IFIN$="("ORIN$=")"THENGOSUB4480
 2510 IFIN$="["ORIN$="]"THENGOSUB4480
 2520 GOTO2390
 2530 WC=WC+1
 2540 GOSUB4580:REM GET PB
 2550 WW=PB+WC
 2560 FORX=0TOWW
 2570 CC=CC+1:IFCC>52THENCC=0
 2580 NEXTX
 2590 IFXL=1703THENXL=1705
 2600 IFXL=1743THENXL=1745
 2610 IFXL=1783THENXL=1785
 2620 IFXL=1823THENXL=1825
```

```
2630 IFXL=1863THENXL=1865
2640 IFXL=1903THENXL=1905
2650 IFXL=1943THENXL=1945
2660 POKEXL,CP(CC)
2670 GOSUB3270:REM WHEELS & SOUND
2680 XL=XL+1:TW=TW+1
2690 IFTW=241THENPOKE1979,19
2700 IFTW=241THENPOKE1980,20
2710 IFTW=241THENPOKE1981,15
2720 IFTW=241THENPOKE1982,16
2730 IFTW=241THEN3090
2740 GOTO2390
2750 REM *** TEST DECRYPT
2760 CC=0:WW=0:XL=1665:TW=1:X=0
2770 POKE1180,81:POKE55452,6
2780 GETIN$:IFIN$=""THEN2780
2790 FORX=0TO52
2800 IFIN$=CC$(X)THENCC=X:GOTO2830
2810 NEXTX
2820 GOTO2780
2830 WC=WC+1
2840 GOSUB4580:REM GET PB
2850 WW=PB+WC
2860 FORX=0TOWW
2870 CC=CC-1:IFCC<0THENCC=52
```

```
2880 NEXTX
2890 IFXL=1703THENXL=1705
2900 IFXL=1743THENXL=1745
2910 IFXL=1783THENXL=1785
2920 IFXL=1823THENXL=1825
2930 IFXL=1863THENXL=1865
2940 IFXL=1903THENXL=1905
2950 IFXL=1943THENXL=1945
2960 POKEXL,CP(CC)
2970 GOSUB3270:REM WHEELS & SOUND
2980 XL=XL+1:TW=TW+1
2990 IFTW=241THENPOKE1979,19
3000 IFTW=241THENPOKE1980,20
3010 IFTW=241THENPOKE1981,15
3020 IFTW=241THENPOKE1982,16
3030 IFTW=241THEN3090
3040 GOTO2780
3050 REM *** INPUT LETTER AND DECRYPT
3060 REM *** MAKE CLICK FOR LETTER
3070 REM *** BUZZER FOR BAD LETTER
3080 END
3090 GOTO5360
3100 DATA@,A,B,C,D,E,F,G,H,I,J,K,L,M
3110 DATAN,O,P,Q,R,S,T,U,V,W,X,Y,Z,"["
3120 DATA"#","]","^","_"," "
```

```
 3130 DATA"!"," ","#","$","%","&","'"
 3140 DATA"(",")","*","+",",","-","."
 3150 DATA"/",0,1,2,3,4,5,6,7,8,9,":"
 3160 DATA";","<","=",">","?"
 3170 DATA@,A,B,C,D,E,F,G,H,I,J,K,L,M
 3180 DATAN,O,P,Q,R,S,T,U,V,W,X,Y,Z
 3190 DATA" ",".","/",0,1,2,3,4,5,6,7
 3200 DATA8,9,":"
 3210
DATA"#","$","%","&","'","-","+","*",",","?","!","="
 3220 DATA0,1,2,3,4,5,6,7,8,9,10,11,12
 3230 DATA13,14,15,16,17,18,19,20,21
 3240 DATA22,23,24,25,26,32,46,47,48
 3250 DATA49,50,51,52,53,54,55,56,57,58
 3260 DATA35,36,37,38,39,45,43,42,44,63,33,61
 3270 REM WHEELS&SOUND
 3280 POKE1198,W4:POKE1202,W5
 3290 W5=W5+1:IFW5=58THENW5=1
 3300 IFW5=27THENW4=W4+1
 3310 IFW5=27THENW5=48
 3320 FORR=54272TO54296:POKER,0:NEXTR
 3330 POKE54296,15:POKE54275,8
 3340 POKE54277,0:POKE54278,240
 3350 POKE54272,47:POKE54273,65
 3360 POKE54276,65:POKE54276,64
```

```
3370 POKE54296,0
3380 REM *** NEED TO LIGHT UP LETTER
3390 IFCC=0THENPOKE1534,128:GOSUB4460
3400 IFCC=0THENPOKE1534,0:RETURN
3410 IFCC=1THENPOKE1436,129:GOSUB4460
3420 IFCC=1THENPOKE1436,1:RETURN
3430 IFCC=2THENPOKE1485,130:GOSUB4460
3440 IFCC=2THENPOKE1485,2:RETURN
3450 IFCC=3THENPOKE1481,131:GOSUB4460
3460 IFCC=3THENPOKE1481,3:RETURN
3470 IFCC=4THENPOKE1440,132:GOSUB4460
3480 IFCC=4THENPOKE1440,4:RETURN
3490 IFCC=5THENPOKE1399,133:GOSUB4460
3500 IFCC=5THENPOKE1399,5:RETURN
3510 IFCC=6THENPOKE1442,134:GOSUB4460
3520 IFCC=6THENPOKE1442,6:RETURN
3530 IFCC=7THENPOKE1444,135:GOSUB4460
3540 IFCC=7THENPOKE1444,7:RETURN
3550 IFCC=8THENPOKE1446,136:GOSUB4460
3560 IFCC=8THENPOKE1446,8:RETURN
3570 IFCC=9THENPOKE1409,137:GOSUB4460
3580 IFCC=9THENPOKE1409,9:RETURN
3590 IFCC=10THENPOKE1448,138:GOSUB4460
3600 IFCC=10THENPOKE1448,10:RETURN
3610 IFCC=11THENPOKE1450,139:GOSUB4460
```

```
3620 IFCC=11THENPOKE1450,11:RETURN
3630 IFCC=12THENPOKE1491,140:GOSUB4460
3640 IFCC=12THENPOKE1491,12:RETURN
3650 IFCC=13THENPOKE1489,141:GOSUB4460
3660 IFCC=13THENPOKE1489,13:RETURN
3670 IFCC=14THENPOKE1487,142:GOSUB4460
3680 IFCC=14THENPOKE1487,14:RETURN
3690 IFCC=15THENPOKE1411,143:GOSUB4460
3700 IFCC=15THENPOKE1411,15:RETURN
3710 IFCC=16THENPOKE1475,144:GOSUB4460
3720 IFCC=16THENPOKE1475,16:RETURN
3730 IFCC=17THENPOKE1395,145:GOSUB4460
3740 IFCC=17THENPOKE1395,17:RETURN
3750 IFCC=18THENPOKE1401,146:GOSUB4460
3760 IFCC=18THENPOKE1401,18:RETURN
3770 IFCC=19THENPOKE1438,147:GOSUB4460
3780 IFCC=19THENPOKE1438,19:RETURN
3790 IFCC=20THENPOKE1403,148:GOSUB4460
3800 IFCC=20THENPOKE1403,20:RETURN
3810 IFCC=21THENPOKE1407,149:GOSUB4460
3820 IFCC=21THENPOKE1407,21:RETURN
3830 IFCC=22THENPOKE1483,150:GOSUB4460
3840 IFCC=22THENPOKE1483,22:RETURN
3850 IFCC=23THENPOKE1397,151:GOSUB4460
3860 IFCC=23THENPOKE1397,23:RETURN
```

```
3870 IFCC=24THENPOKE1479,152:GOSUB4460
3880 IFCC=24THENPOKE1479,24:RETURN
3890 IFCC=25THENPOKE1477,153:GOSUB4460
3900 IFCC=25THENPOKE1477,25:RETURN
3910 IFCC=26THENPOKE1405,154:GOSUB4460
3920 IFCC=26THENPOKE1405,26:RETURN
3930 IFCC=27THENPOKE1484,160:GOSUB4460
3940 IFCC=27THENPOKE1484,32:RETURN
3950 IFCC=28THENPOKE1413,174:GOSUB4460
3960 IFCC=28THENPOKE1413,46:RETURN
3970 IFCC=29THENPOKE1452,175:GOSUB4460
3980 IFCC=29THENPOKE1452,47:RETURN
3990 IFCC=30THENPOKE1514,176:GOSUB4460
4000 IFCC=30THENPOKE1514,48:RETURN
4010 IFCC=31THENPOKE1516,177:GOSUB4460
4020 IFCC=31THENPOKE1516,49:RETURN
4030 IFCC=32THENPOKE1518,178:GOSUB4460
4040 IFCC=32THENPOKE1518,50:RETURN
4050 IFCC=33THENPOKE1520,179:GOSUB4460
4060 IFCC=33THENPOKE1520,51:RETURN
4070 IFCC=34THENPOKE1522,180:GOSUB4460
4080 IFCC=34THENPOKE1522,52:RETURN
4090 IFCC=35THENPOKE1524,181:GOSUB4460
4100 IFCC=35THENPOKE1524,53:RETURN
4110 IFCC=36THENPOKE1526,182:GOSUB4460
```

```
4120 IFCC=36THENPOKE1526,54:RETURN
4130 IFCC=37THENPOKE1528,183:GOSUB4460
4140 IFCC=37THENPOKE1528,55:RETURN
4150 IFCC=38THENPOKE1530,184:GOSUB4460
4160 IFCC=38THENPOKE1530,56:RETURN
4170 IFCC=39THENPOKE1532,185:GOSUB4460
4180 IFCC=39THENPOKE1532,57:RETURN
4190 IFCC=40THENPOKE1493,186:GOSUB4460
4200 IFCC=40THENPOKE1493,58:RETURN
4210 IFCC=41THENPOKE1420,163:GOSUB4460
4220 IFCC=41THENPOKE1420,35:RETURN
4230 IFCC=42THENPOKE1421,164:GOSUB4460
4240 IFCC=42THENPOKE1421,36:RETURN
4250 IFCC=43THENPOKE1422,165:GOSUB4460
4260 IFCC=43THENPOKE1422,37:RETURN
4270 IFCC=44THENPOKE1460,166:GOSUB4460
4280 IFCC=44THENPOKE1460,38:RETURN
4290 IFCC=45THENPOKE1461,167:GOSUB4460
4300 IFCC=45THENPOKE1461,39:RETURN
4310 IFCC=46THENPOKE1462,173:GOSUB4460
4320 IFCC=46THENPOKE1462,45:RETURN
4330 IFCC=47THENPOKE1500,171:GOSUB4460
4340 IFCC=47THENPOKE1500,43:RETURN
4350 IFCC=48THENPOKE1501,170:GOSUB4460
4360 IFCC=48THENPOKE1501,42:RETURN
```

```
4370 IFCC=49THENPOKE1502,172:GOSUB4460
4380 IFCC=49THENPOKE1502,44:RETURN
4390 IFCC=50THENPOKE1540,189:GOSUB4460
4400 IFCC=50THENPOKE1540,63:RETURN
4410 IFCC=51THENPOKE1541,161:GOSUB4460
4420 IFCC=51THENPOKE1541,33:RETURN
4430 IFCC=52THENPOKE1542,187:GOSUB4460
4440 IFCC=52THENPOKE1542,61:RETURN
4450 RETURN
4460 FORZ=1TO100:NEXTZ
4470 RETURN
4480 REM BUZZER SOUND
4490 POKE1176,81:POKE55448,2
4500 FORS1=1TO12
4510 POKE54296,15
4520 FORS2=1TO5:NEXT
4530 POKE54296,0
4540 FORS3=1TO5:NEXT
4550 NEXT
4560 POKE1176,81:POKE55448,12
4570 RETURN
4580 REM PB CODE
4590 P1=P1+1:IFP1>4THENP1=1
4600 IFP1$="A"ANDP1=1THENPB=1
4610 IFP1$="A"ANDP1=2THENPB=4
```

```
4620 IFP1$="A"ANDP1=3THENPB=8
4630 IFP1$="A"ANDP1=4THENPB=12
4640 IFP1$="B"ANDP1=1THENPB=3
4650 IFP1$="B"ANDP1=2THENPB=6
4660 IFP1$="B"ANDP1=3THENPB=9
4670 IFP1$="B"ANDP1=4THENPB=13
4680 IFP1$="C"ANDP1=1THENPB=5
4690 IFP1$="C"ANDP1=2THENPB=10
4700 IFP1$="C"ANDP1=3THENPB=15
4710 IFP1$="C"ANDP1=4THENPB=20
4720 IFP1$="D"ANDP1=1THENPB=6
4730 IFP1$="D"ANDP1=2THENPB=11
4740 IFP1$="D"ANDP1=3THENPB=18
4750 IFP1$="D"ANDP1=4THENPB=24
4760 IFP2=1THENPB=PB+3
4770 IFP2=2THENPB=PB+7
4780 IFP2=3THENPB=PB+9
4790 IFP2=4THENPB=PB+11
4800 RETURN
4810 REM * NUMBER TO POKE CODE
4820 A1=0
4830 IFA$="A"THEN A1=1
4840 IFA$="B"THEN A1=2
4850 IFA$="C"THEN A1=3
4860 IFA$="D"THEN A1=4
```

```
4870 IFA$="E"THEN A1=5
4880 IFA$="F"THEN A1=6
4890 IFA$="G"THEN A1=7
4900 IFA$="H"THEN A1=8
4910 IFA$="I"THEN A1=9
4920 IFA$="J"THEN A1=10
4930 IFA$="K"THEN A1=11
4940 IFA$="L"THEN A1=12
4950 IFA$="M"THEN A1=13
4960 IFA$="N"THEN A1=14
4970 IFA$="O"THEN A1=15
4980 IFA$="P"THEN A1=16
4990 IFA$="Q"THEN A1=17
5000 IFA$="R"THEN A1=18
5010 IFA$="S"THEN A1=19
5020 IFA$="T"THEN A1=20
5030 IFA$="U"THEN A1=21
5040 IFA$="V"THEN A1=22
5050 IFA$="W"THEN A1=23
5060 IFA$="X"THEN A1=24
5070 IFA$="Y"THEN A1=25
5080 IFA$="Z"THEN A1=26
5090 IFA$="0"THEN A1=48
5100 IFA$="1"THEN A1=49
5110 IFA$="2"THEN A1=50
```

```
5120 IFA$="3"THEN A1=51
5130 IFA$="4"THEN A1=52
5140 IFA$="5"THEN A1=53
5150 IFA$="6"THEN A1=54
5160 IFA$="7"THEN A1=55
5170 IFA$="8"THEN A1=56
5180 IFA$="9"THEN A1=57
5190 IFA$="@"THEN A1=0
5200 IFA$="."THEN A1=46
5210 IFA$="/"THEN A1=47
5220 IFA$=":"THEN A1=58
5230 IFA$="#"THEN A1=35
5240 IFA$="$"THEN A1=36
5250 IFA$="%"THEN A1=37
5260 IFA$="&"THEN A1=38
5270 IFA$="'"THEN A1=39
5280 IFA$="-"THEN A1=45
5290 IFA$="+"THEN A1=43
5300 IFA$="*"THEN A1=42
5310 IFA$=","THEN A1=44
5320 IFA$="?"THEN A1=63
5330 IFA$="!"THEN A1=33
5340 IFA$="="THEN A1=61
5350 RETURN
5360 FORX=0TO37:Y=PEEK(1665+X):GOSUB5490:NEXTX
```

```
5370 FORX=0TO37:Y=PEEK(1705+X):GOSUB5490:NEXTX
5380 POKE55897,2:POKE1625,53
5390 FORX=0TO37:Y=PEEK(1745+X):GOSUB5490:NEXTX
5400 POKE55898,2:POKE1626,52
5410 FORX=0TO37:Y=PEEK(1785+X):GOSUB5490:NEXTX
5420 POKE55899,2:POKE1627,51
5430 FORX=0TO37:Y=PEEK(1825+X):GOSUB5490:NEXTX
5440 POKE55900,2:POKE1628,50
5450 FORX=0TO37:Y=PEEK(1865+X):GOSUB5490:NEXTX
5460 POKE55901,2:POKE1629,49
5470 FORX=0TO26:Y=PEEK(1905+X):GOSUB5490:NEXTX
5480 GOTO6030
5490 IFY=35THENME$(CO)="#":CO=CO+1:RETURN
5500 IFY=36THENME$(CO)="$":CO=CO+1:RETURN
5510 IFY=37THENME$(CO)="%":CO=CO+1:RETURN
5520 IFY=38THENME$(CO)="&":CO=CO+1:RETURN
5530 IFY=39THENME$(CO)="'":CO=CO+1:RETURN
5540 IFY=45THENME$(CO)="-":CO=CO+1:RETURN
5550 IFY=43THENME$(CO)="+":CO=CO+1:RETURN
5560 IFY=42THENME$(CO)="*":CO=CO+1:RETURN
5570 IFY=44THENME$(CO)=",":CO=CO+1:RETURN
5580 IFY=63THENME$(CO)="?":CO=CO+1:RETURN
5590 IFY=33THENME$(CO)="!":CO=CO+1:RETURN
5600 IFY=61THENME$(CO)="=":CO=CO+1:RETURN
5610 IFY=17THENME$(CO)="Q":CO=CO+1:RETURN
```

```
5620 IFY=23THENME$(CO)="W":CO=CO+1:RETURN
5630 IFY=5 THENME$(CO)="E":CO=CO+1:RETURN
5640 IFY=18THENME$(CO)="R":CO=CO+1:RETURN
5650 IFY=20THENME$(CO)="T":CO=CO+1:RETURN
5660 IFY=26THENME$(CO)="Z":CO=CO+1:RETURN
5670 IFY=21THENME$(CO)="U":CO=CO+1:RETURN
5680 IFY=9 THENME$(CO)="I":CO=CO+1:RETURN
5690 IFY=15THENME$(CO)="O":CO=CO+1:RETURN
5700 IFY=46THENME$(CO)=".":CO=CO+1:RETURN
5710 IFY=1 THENME$(CO)="A":CO=CO+1:RETURN
5720 IFY=32THENME$(CO)=" ":CO=CO+1:RETURN
5730 IFY=19THENME$(CO)="S":CO=CO+1:RETURN
5740 IFY=4 THENME$(CO)="D":CO=CO+1:RETURN
5750 IFY=6 THENME$(CO)="F":CO=CO+1:RETURN
5760 IFY=7 THENME$(CO)="G":CO=CO+1:RETURN
5770 IFY=8 THENME$(CO)="H":CO=CO+1:RETURN
5780 IFY=10THENME$(CO)="J":CO=CO+1:RETURN
5790 IFY=11THENME$(CO)="K":CO=CO+1:RETURN
5800 IFY=47THENME$(CO)="/":CO=CO+1:RETURN
5810 IFY=16THENME$(CO)="P":CO=CO+1:RETURN
5820 IFY=25THENME$(CO)="Y":CO=CO+1:RETURN
5830 IFY=24THENME$(CO)="X":CO=CO+1:RETURN
5840 IFY=3 THENME$(CO)="C":CO=CO+1:RETURN
5850 IFY=22THENME$(CO)="V":CO=CO+1:RETURN
5860 IFY=2 THENME$(CO)="B":CO=CO+1:RETURN
```

```
5870 IFY=14THENME$(CO)="N":CO=CO+1:RETURN
5880 IFY=13THENME$(CO)="M":CO=CO+1:RETURN
5890 IFY=12THENME$(CO)="L":CO=CO+1:RETURN
5900 IFY=58THENME$(CO)=":":CO=CO+1:RETURN
5910 IFY=48THENME$(CO)="0":CO=CO+1:RETURN
5920 IFY=49THENME$(CO)="1":CO=CO+1:RETURN
5930 IFY=50THENME$(CO)="2":CO=CO+1:RETURN
5940 IFY=51THENME$(CO)="3":CO=CO+1:RETURN
5950 IFY=52THENME$(CO)="4":CO=CO+1:RETURN
5960 IFY=53THENME$(CO)="5":CO=CO+1:RETURN
5970 IFY=54THENME$(CO)="6":CO=CO+1:RETURN
5980 IFY=55THENME$(CO)="7":CO=CO+1:RETURN
5990 IFY=56THENME$(CO)="8":CO=CO+1:RETURN
6000 IFY=57THENME$(CO)="9":CO=CO+1:RETURN
6010 IFY=0 THENME$(CO)="@":CO=CO+1:RETURN
6020 RETURN
6030 OPEN4,4:CMD4
6040 FORX=0TOCO
6050 PRINT ME$(X);
6060 NEXTX
6070 PRINT#4
6080 CLOSE4
6090 END
```

Keyfinder

This is a little program I wrote to help me find the characters I needed when running the 64.EMU emulator on Android.

```
 10 GETA$:IFA$=""THEN10
 20 IFA$=CHR$(5)THENPRINT"WHITE":GOTO10
 30 IFA$=CHR$(13)THENPRINT"RETURN":GOTO10
 40 IFA$=CHR$(17)THENPRINT"CRSR DN":GOTO10
 50 IFA$=CHR$(18)THENPRINT"RVS ON":GOTO10
 60 IFA$=CHR$(19)THENPRINT"CLR/HOME":GOTO10
 70 IFA$=CHR$(28)THENPRINT"RED":GOTO10
 80 IFA$=CHR$(29)THENPRINT"CRSR RIGHT":GOTO10
 90 IFA$=CHR$(30)THENPRINT"GREEN":GOTO10
100 IFA$=CHR$(31)THENPRINT"BLUE":GOTO10
110 IFA$=CHR$(32)THENPRINT"SPACE":GOTO10
120 IFA$=CHR$(142)THENPRINT"UPPER CASE":GOTO10
130 IFA$=CHR$(144)THENPRINT"BLACK":GOTO10
140 IFA$=CHR$(145)THENPRINT"CRSR UP":GOTO10
150 IFA$=CHR$(146)THENPRINT"RVS OFF":GOTO10
160 IFA$=CHR$(147)THENPRINT"CLR/HOME":GOTO10
170 IFA$=CHR$(148)THENPRINT"INST/DEL":GOTO10
180 IFA$=CHR$(156)THENPRINT"PURPLE":GOTO10
190 IFA$=CHR$(157)THENPRINT"CRSR LEFT":GOTO10
200 IFA$=CHR$(158)THENPRINT"YELLOW":GOTO10
210 IFA$=CHR$(159)THENPRINT"CYAN":GOTO10
220 IFA$=CHR$(160)THENPRINT"SPACE":GOTO10
230 PRINTA$:GOTO10
```

Errata

In my previous book on page 13, I have CHR$(145)=Cursor UP but it should read Cursor DN.